

MICROCOPY RESOLUTION TEST CHART NATIONAL BUREAU OF STANDARDS 1963 A

AD A 1 2 9 1 8 2

H

SECURITY CEASSIFICATION OF THIS PAGE (WHEN DATE BRITANE)							
REPORT DOCUMENTATI	READ INSTRUCTIONS BEFORE COMPLETING FORM						
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER					
UAA002	AD. 4129 1	jr_					
4. TITLE (fund Subtitle)	•	5. TYPE OF REPORT & PERIOD COVERED					
U.S. ARMY INTELLIGENCE CENTER							
USAICS, Software Analysis and		FINAL					
System Analysis of Geographic Algorithms.	Transformation	6. PERFORMING ORG. REPORT NUMBER					
Algorithms. 7. Author(e)		8. CONTRACT OR GRANT NUMBER(a)					
JET PROPULSION LABORATORY, Cal Institute of Technology,	ifornia						
Pasadena California		NAS7- 918					
9. PERFORMING ORGANIZATION NAME AND ADD	RESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS					
Jet Propulsion Laboratory	ATTN: 171-209	AREA & WORK UNIT NUMBERS					
California Institute of Techno	ology	RE 182					
4800 Oak Grove, Pasadena, CA		AMEND # 187					
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE					
Commander		JULY 9, 1982					
USAICS ATTN: ATSI-C	CD-SF	13. NUMBER OF PAGES					
Ft Huachuca, AZ 85613		35					
14. MONITORING AGENCY NAME & ADDRESS(IL di	Herent from Controlling Office)	15. SECURITY CLASS. (of thie report)					
(Same as ITEM 11)		UNCLAS					
Commander, USAICS	•	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NONE					
16. DISTRIBUTION STATEMENT (of this Report)							

Approved for Public dissemination

17. DISTRIBUTION STATEMENT (of the abetract entered in Block 20, If different from Report)

18. SUPPLEMENTARY NOTES

Prepared by the Jet Propulsion Labs for the US Army Intelligence Center and School.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

GEOGRAPHIC TRANSFORMATIC ALGORITHM.

20. ABSTRACT (Coulines on reverse shift if neserously and identify by block number)

This report describes the findings of JPL regarding geographic transformation algorithms used in MAGIIC, Guardrail, Trailblazer and BETA systems. A set of parameters is developed to characterize and catalogue intelligence system algorithms in the 4 systems. Individual algorithms are also analyzed to determine if they are performing their functions properly.

DD 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OSSOLETE

UNCLAS

U.S. ARMY INTELLIGENCE CENTER AND SCHOOL USAICS

Software Analysis and Management System

Analysis of Geographic Transformation Algorithms

July 9, 1982

Approval:

H. F. Lesh. Manager

USAICS Software Analysis and

Management System

John Radbill, Task Leader

Algorithm Analysis

USAICS Software Analysis and

Management System

H. Jeane, Manager

Software Development Section

A. Ferrari. Manager

Defense Information Systems Program

Concur:

Major W. Akins, Chief

All Source Analysis System Office

U.S. Army Intelligence Center and School

SELECTE JUN 9 1983

This document has been approved for public release and sale; its distribution is unlimited.

JET PROPULSION LABORATORY
California Institute of Technology
Pasadena, California

The following people contributed to this report

- P. Babby
- J. Gillis
- A. Griesel
- N. Palmer
- J. Radbill



Accession For
NTIS GRAEI
DTIC TAB
15 WAS A SECOND
Justification
By
Distribution/
Avoilability Codes
Aveil mid/or
Disco Special
1774
141
1111

TABLE OF CONTENTS

1.	Introduction	1
	1.1. Purpose	1
	1.2. Background	1
	1.3. User Benefits	3
2.	Analyzing the Algorithms	4
	2.1. Early Steps	4
	2.2. Learning Military Mapping	4
	2.3. Representing Algorithms in Standard Form	7
3.	Characterizing and Cataloging the Algorithms	12
4.	Algorithms in MAGIIC	14
	4.1. MAGIIC Lambert Constant Generation Algorithm	14
	4.2. An Interrelated Set of MAGIIC Coordinate Conversion Algorithms	14
	4.3. MAGIIC Polar Grid to Universal Polar Sterographic Algorithm	16
	4.4. MAGIIC Northing and Easting to Universal Transverse Mercator Algorithm	17
5.	Comparing the Algorithms Across Systems	19
	5.1. Spheroid Models	19
	5.2. Grid Zone Generation	19
	5.3. MAGIIC Grid Zone Generation Algorithm	20
	5.4. Guardrail Grid Zone Generation Algorithm	20
	5.5. Trailblazer Grid Zone Generation Algorithm	21
	5.6. BETA Grid Zone Generation Algorithm	21
6.	Discussion and Conclusions	24
	6.1. Documentation of Algorithms	24
	6.2. Similarity of Functions Across Systems	24
	6.3. Incompleteness of Algorithms for Global Applications	24
	6.4. Robustness of Algorithms	24
	6.5. Selection/Consolidation of Algorithms	25

7. Appendi	loes	•••••	• • •	• • •	• • •	•••	•••	••	• • •	••	••	••	••	••	• • •	•	• • •	• •	•••	• •	••	•••	•••	. 27
7.1. Da	tabase Ent	ries and	Pr	odi	uet	s ()	PSL	/P:	SA)			• • •		• • •		• •	• • •		• •					. 28
7.2. Al	lgorithm Hi	erarchy (ha	rt:	з	• • •			• • •		• •			••			• • •		• • •			• • •	• • •	. 29
7.3. Al	lgorithms i	n Standaı	rd	For	m.	• • •	•••	• •		••	• •	• • •	••	• • •	• • •	••	• • •	•	••	• ‹	••	• • •	• .	. 30
	-																							
		L	I	8	ľ	0	F	F	I	G	Ū	R	E	S										
		_												_										
2-1: Grid	Zone Desig	nations o	of	th	e M	ili	tar	y (Gri	.d	Re	fe	re	nc	e S	Sys	ter	n (UI	M)	٠.	• • •		. 9
	Zone Desig																							
	Plan of t												_	_										
						-,										• •								
2_1. Algor	rithm Level	Weln Wil	ه ا																					. 13
3-1. KIEW	TOUR HOVEL	Herb LT	. • •	• • •	• • •	•••	•••	••	• • •	• •	••	••	••	••	• • •	• •	•••	• •	• •	••	•••	•••	•••	• 13
6 1. Decu	mentation o	e Algori	t has									٠												. 26
0-1: Docur	mentarion c	or WIROLI	ш	13.	• • •	•••		•••	• • •	• • •	• •	••	••	••	• • •	• • •	••	•	• • •	• •	• • •	•••	•••	• 20
		1	LI	S	T	0	F		T A	E	L	. B	S											
		•										_	_											
5-1: Inco	nsistent Sp	heroid U	38. 8	ζe.	• • •	•••	• • •	••	•••	• •	• •	••	••	••	• • •	•••	• • •	• • •	•••	• •	• • •	•••	•••	. 23

1. Introduction

1.1. Purpose

This report describes the findings of the Algorithm Analysis Subtask group working on the U.S. Army Intelligence Center and School (USAICS) Software Analysis and Management System task (USAMS) regarding geographic transformation algorithms used in four of the intelligence-gathering systems under USAICS cognizance. In this report a set of parameters is developed to characterize and catalogue intelligence system algorithms in four specific systems. Individual algorithms are analyzed to determine whether they are performing their functions properly. Algorithms that perform the same function in different systems are compared to determine which ones are best according to various criteria.

The algorithms examined in this report are taken from the MAGIIC, Guardrail, Trailblazer, and BETA systems. They were chosen from the approximately 41 deployed intelligence systems for which USAICS is Combat Developer because their documentation was quickly accessible and because they represented a range of algorithm applications. Geographic transformation (mapping) algorithms were chosen for this report since all four systems contain position location/description functions and many of their algorithms are unclassified.

1.2. Background

Each of the about 41 intelligence systems under USAICS cognizance employs several types of algorithms to carry out its gathering and processing of intelligence data. Two important types of these algorithms, geographic transformation and correlation, have been chosen for analysis during this year. The former translates grid zone locations, for example, from latitude-longitude to Universal Transverse Mercator (UTM), while the latter resolves many individual sitings into militarily recognizable targets based chiefly on standard statistical procedures. It is important to develop a set of parameters to characterize these algorithms so that how they should be catalogued can be

Two additional reports will be submitted in FY82: a correlation analysis report and a report on possible algorithm analysis methodologies.

determined. When these activities are completed, it becomes possible to compare algorithms that perform the same function in different systems.

To begin this process, the JPL Algorithm Analysis Sub-task group has examined the geographic transformation algorithms for four of the 41 systems, namely The Mobile Army Ground Imagery Interpretation Center (MAGIIC), Guardrail, Trailblazer, and Battlefield Exploitation and Target Acquisition (BETA). These four systems chosen for more detailed study represent several intelligence data analysis functions. MAGIIC is a ground-based analysis system to assist in interpreting hard-copy images from different airborne surveillance systems, including a capability for computerized mensuration on imagery; it can also receive and analyze data from Tactical Electronic Intelligence (TEREC) collection systems and provide emitter location estimates. Guardrail uses airborne sensor platforms to collect data on Direction Finding (DF) emitters; extensive ground-based software is then used to estimate the location of the units, such as command posts, associated with these emitters. Trailblazer also uses DF data to estimate emitter location. Its sensor platforms are essentially fixed and ground-based. BETA is a Test Bed program for correlating data received from several types of sensor systems and making target nominations. Both automatic correlation and aggregation techniques and interactive graphics are used in the operator's analysis. These systems would generally be employed at Division or Corps level or at an Air Force Tactical Air Control Element (TACE) or Allied Tactical Air Force (ATAF); target nominations and tactical situation reports would be available to commanders and their staffs from Brigade through Echelons Above Corps (EAC).

USAICS has cognizance of a large number of algorithms integral to intelligence-gathering systems in various stages of development and deployment. The state of "deployment" of algorithms in the USAICS inventory ranges from that of products of research contracts not yet implemented in any system to those in fielded systems such as Trailblazer or Guardrail. In the latter systems the algorithms are documented in design documents (narrative English and equations), and/or in machine readable design language, and in code. Often not all of these forms of documentation are available for any one system. For research algorithms not yet implemented actual code, or even detailed flow charts, may not be available; thus analysis must rely solely on mathematical descriptions.

"Algorithm" means any set of rules for carrying out a single conceptual operation on a set of data, such as transforming latitude-longitude coordinates to UTM or determining a position from a number of direction measurements taken at known points. Algorithms are often hierarchical, lower-level algorithms often being used to describe higher-level algorithms and thereby illuminating their underlying logical structure. Thus, results from one algorithm may be data for another. USAICS is interested in algorithms performing intelligence data processing functions central to their systems' mission and those performing crucial support functions, such as geographic location, common to a number of systems. Data management or mathematical function algorithms, although vital to the efficient functioning of the systems, are not being treated in these first algorithm analyses.

1.3. User Benefits

These analyses can benefit users in several ways. First, a catalog of existing algorithms will help USAICS avoid having algorithms redeveloped for new systems from first principles. Second, analysis of individual algorithms may, in a few cases, identify deficiencies worth correcting on the next system revision. Third, and most important, the comparison of algorithms performing the same function in different systems can lead to identifying guidelines for developing and/or selecting algorithms to include in new and revised systems. Selected algorithms from the systems studied will begin to form a library of intelligence algorithms with associated computer subroutines that will be analogous to the Collected Algorithms of the Association for Computing Machinery (ACM). The creation of such a library is in the spirit of Ada⁺, the Department of Defense language for embedded systems, and Ada's environment.

These conceptual models should be describable, although their technical implementation is often significantly more complicated to present.

⁺Ada is a trademark of the Department of Defense

2. Analyzing the Algorithms

2.1. Early Steps

Since the Location and Movement Analysis System (LAMAS) system documentation was available first, our early analysis efforts were directed to that system. A preliminary analysis of a Shortest Path Algorithm was done and modeled in Pascal as an approach to standardizing representations. This algorithm was a variant of Dijkstra's Shortest Path Algorithm.

Later the sponsor decided that our first emphasis should be on the coordinate conversion algorithms of the MAGIIC, Guardrail, Trailblazer and BETA systems. Three approaches to this analysis were tried and evaluated. The MAGIIC system has been hierarchically analyzed for the interrelationship of the algorithms. The spheroid models of the Earth's oblateness have been examined for all the systems. The grid zone generation algorithms have been compared across all four systems.

2.2. Learning Military Mapping

To analyze the first type of algorithm required learning the military grid system. This discussion identifies the various map projections and military grid reference systems examined and how they are interrelated. The scope of this discussion is limited to only those map projections and grid reference systems pertinent to the MAGIIC, BETA, Guardrail, and Trailblazer systems.

The map projections discussed are the Transverse Mercator, Polar Stereographic, Lambert Conformal Conic, and the Gnomonic. The grid reference systems used are the Universal Transverse Mercator (UTM), Universal Polar Stereographic (UPS), and Military Grid Reference System (MGR). The selection of a map projection is based on the properties it preserves in the transformation from a three-dimensional spheroid to a two-dimensional plane. These properties include orthogonality of latitude and longitude, equal area representation, distortion of shape, minimal change in scale factor in either eastwest or north-south directions, and representation of great circles by straight lines. Since all map projections are from a spheroid model of the Earth, the parameters that the spheroid model use are very important. Various spheroid models are used for different portions of the Earth.

The selection of a grid reference system depends on the portion of the Earth examined and the resolution desired. The UTM and UPS coordinate systems were adopted as standards by the military to minimize coordination problems due to the proliferation of locally-used grid reference systems. These coordinate systems are most suitable for the representation of large geographic areas (greater than 9° in latitude and longitude). The MGR system provides greater resolution when representing smaller geographic areas (within 100,000-meter by 100,000-meter squares). The MGR system can be overlaid on the UTM and UPS coordinate systems to eliminate ambiguity due to repetitions of the 100,000-meter square identifiers. The geographic reference system is simply given as a longitude-latitude pair. However, this reference system of zones is cumbersome for representing locations in good resolution. Also, there is an inconsistency in the form of the coordinates: some applications use decimal degree notation while others use clock-like representations.

The UTM grid reference system is valid for all longitudes over latitudes between 84° North and 80° South. This area is divided into rectangles of 6° in longitude (zones) by 8° in latitude (bands), except for the 12° band from 72° to 84° latitude. There are 60 zones numbered from 1 through 60 for the zones from -180° to +180° longitude. There are 22 bands lettered C through X for the bands from -80° to 84° latitude. There are some subtle irregularities in this pattern beyond 56° latitude between 0° and 45° in longitude (see Figure 2-1). The UTM grid reference system is based on the Modified Transverse Mercator projection, but can be mathematically transformed for use with other types of projections.

The UPS grid reference system is valid for the North Polar (+84° longitude to the pole) and the South Polar regions (-80° longitude to the pole). These regions have a grid zone number of zero and consist only of a grid zone letter that is longitude-dependent. The North Polar region grid zone letters are Y (Western hemisphere) and Z (Eastern Hemisphere). The South Polar regions are A and B. The UPS grid reference system is based on the Polar Stereographic projection (see Figure 2-2).

The MGR grid reference system, illustrated for the UPS system in Figure 2-2 and for the UTM system in Figure 2-3, provides finer resolution than the UTM or UPS grid reference systems. It identifies 100,000-meter by 100,000-

meter squares by two letters, an Easting letter and a Northing letter. These letters are sequenced so as to provide at least 18° separation between similarily-lettered squares (within a given spheroid model area - otherwise the separation is 9°). These lettering sequences are biased and restarted at the boundaries of the underlying spheroid models.

These MGR system letter designations may be used without reference to the UTM or UPS designations when there is no likelihood of ambiguity, otherwise the UTM or UPS designation is included. Positions within these squares can be interpolated in tens of meters and are referred to as Easting and Northing terms representing distances rather than degrees.

The Transverse Mercator projection transforms the Earth's spheroid onto a cylinder secant to the Earth and perpendicular to its axis. This projection is used at latitudes within 84° North and 80° South. Scale linearity is correct at the two meridians cut by the cylinder (6° apart) and quite accurate in the band formed by them. Because of the vertical linearity this projection is particularly suitable to areas of interest in the North-South direction. This projection lends itself well to being overlaid with a rectangular grid reference system (such as the MGR system).

The Polar Stereographic projection transforms the Earth's spheroid onto a plane tangential to the Earth (at the pole, in our applications). This projection is used at latitudes beyond 84° North and from 80° South. Scale linearity decreases and equal area exaggeration increases as the distance from the pole increases. Latitude-longitude orthogonality is preserved at the meridian crossings. All circles of latitude are concentric, centered at the pole. Thus, this projection is useful for plotting radio waves and air navigation with a compass.

The Lambert Conformal Conic projection transforms the Earth's spheroid onto a cone parallel to the Earth's axis and secant to the earth at two latitudes referred to as the standard latitudes. The East-West scale linearity is correct at the two standard latitudes and is relatively accurate in the band between these latitudes. The projection preserves direction and shape quite well within and near the standard latitudes. Hence, the Lambert, Conformal Conic Projection is best suited to East-West measurements and is useful for air navigation. Also, all meridians are straight and intersect at the pole. This

projection is most applicable to the mid-latitude region where the cone is secant to the Earth.

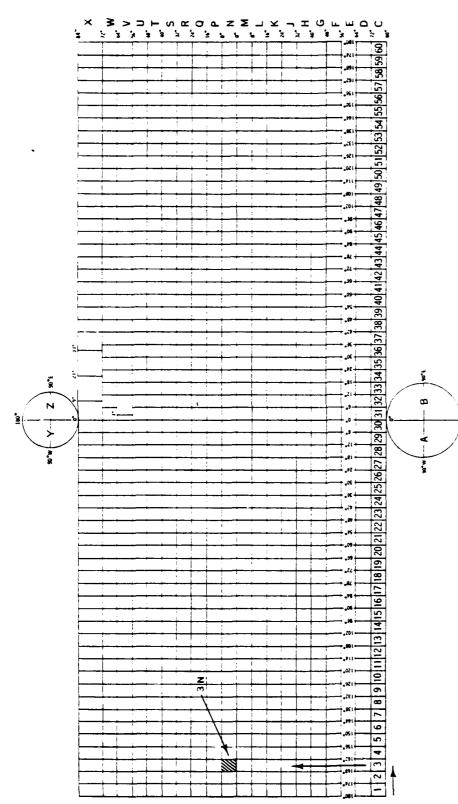
The Gnomonic projection transforms the Earth's spheroid onto a plane tangent to the Earth's at the point of interest. This projection is valid over all latitudes and longitudes. It has the quality of representing all great circle arcs on the projection as straight lines. Since electromagnetic waves travel the shortest distance route (great circle arc), the Gnomonic projection is ideally suited for the presentation of direction-finding lines of bearing.

2.3. Representing Algorithms in Standard Form

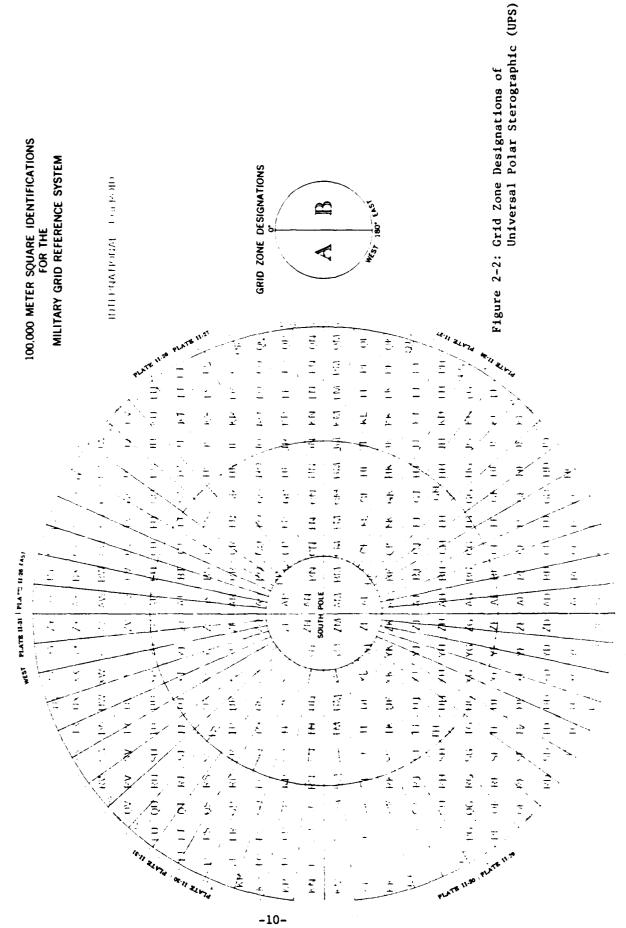
To compare algorithms across systems, all algorithms analyzed must be translated into a standard format. Algorithms in the systems analyzed had been coded in such diverse languages as assembly and structured FORTRAN so that translation into a common Higher-Order Language (HOL) became essential to searching for common and diverse features. Publication ALGOL was seen as an attractive candidate because it has been used in the collected algorithms of the ACM for algorithm description. However, audiences outside the Applied Mathematics, Numerical Analysis, and Computer Sciences communities are generally unfamiliar with ALGOL; and compilers for ALGOL 60 or ALGOL 68 are not readily available in this country.

Pascal, an ALGOL-like language, has been chosen for the primary representation language for the algorithms because it has many of the properties of ALGOL (structure, strong typing, etc.), it has become familiar to a wide audience, and high quality compilers are available on many computers including the Digital Equipment Corp. VAX and many microcomputers with CPM operating systems. The last point is important because the VAX will be used by both USAICS and JPL, and the microcomputers are similar to the word processors and personal computers at JPL and USAICS. Among the features of Pascal that contribute to its clarity are the command structures, such as "if-then-else" and "case", and the user-defined data types. However, separate compilations of procedures to support hierarchical descriptions of algorithms are an implementation-dependent extension rather than a basic feature of the language. Because of this and other problems Pascal provides at best an interim solution to the algorithm description problem.

Ada offers a long-term solution. The Ada language avoids many of the shortcomings of Pascal and has many additional features. A stronger reason for using Ada is that the Army is likely to require all new systems initiated after 1984 to be programmed in it. While no complete compiler for Ada is currently available, there is an interpreter on the project VAX computer, although it is very slow. A compiler for an incomplete implementation available on Z80-based microcomputers with CPM operating systems is also available. Although this compiler is not completely satisfactory because of the lack of user-defined types, it is still useful for some simple examples and for comparison with Pascal.



Grid Zone Designations of the Military Grid Reference System (UTM)



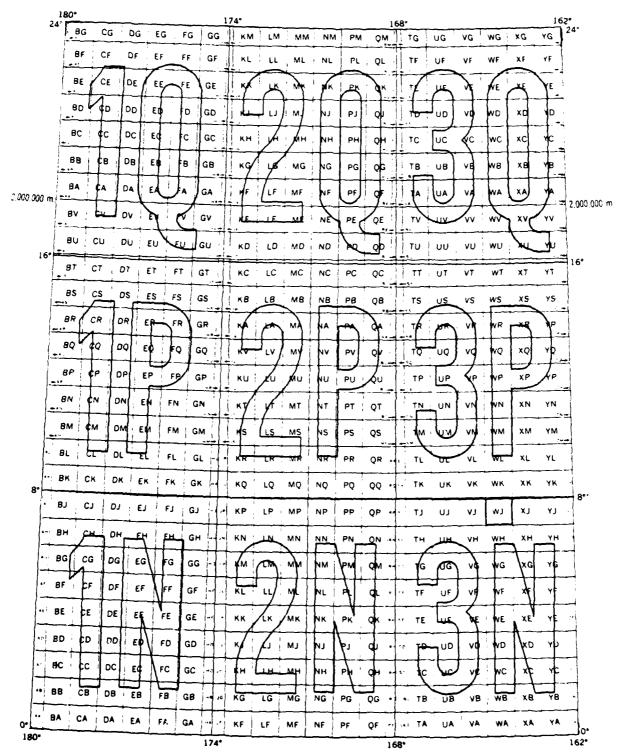


Figure 2-3: Basic Plan of the 100,000-meter Square Identifications of the U.S. Army Military Grid Reference System, between 84° N. and 80° S

3. Characterizing and Cataloging the Algorithms

The set of properties stored for algorithms in the database should characterize an algorithm for military application purposes without requiring that the algorithm itself be retrieved and examined. The algorithm property selection is influenced by the following ways the database will be used. user may wish a general summary of what algorithms are in the database. A more likely use is to look for algorithms that perform a specified function, such as position location. In another dimension, the user may ask "What algorithms do we have in MAGIIC?". Cataloging properties should be independent for efficiency of description. Two of the properties chosen, performance and robustness, are not totally independent. Requirements performance is interpreted here as, "does the algorithm do what it says it does?". Lack of robustness is interpreted here as failure for special value or failure because of small errors in input data. Range checking of input variables is an important contributor to robustness. This is particularly important if the algorithm is to be used for more than one system where the calling programs can not be expected to protect it.

The Algorithm Level Help File; (Figure 3-1) taken from the Acquisition and Database Entry Prompting Tool (ADEPT) User's Guide, is examined and interpreted as a means for describing the present classification scheme. This information is provided for each system in which an algorithm is implemented. This set of parameters is likely to change as more experience is gained with its use.

Examples of PSA reports are shown in Appendix 7.1.

Fig. 3-1: Algorithm Level Help File

- NAME is a name description of the algorithms function.
- SYNONYM is the algorithms abbreviation (the same as its VAX file element name).
- SOURCE: AUTHOR is the document from which the algorithm was taken and author, if known.
- PROCESSING is what JPL has done with the algorithm, e.g. Pascal program tested.
- MATH: FIELD is what mathematical field the algorithm is based on, e.g. least squares.
- ROBUSTNESS is a measure of sensitivity to values of variables input to the algorithm, e.g. a transformation algorithm may produce an incorrect result with inputs of + 180° longitude.
- TREE:LEVEL is a rough indication of location of the algorithm in the hierarchy of algorithms in a system.
- REQUIREMENTS/PERFORMANCE Since the requirements documents for particular algorithms are generally not available, requirements must be derived from design documents or comments in code.

 This item describes how well the algorithm meets these requirements.
- REFERENCE is a pointer to the VAX file containing the representation of the algorithm in standard form: Pascal and in some cases Ada. (These are given in Appendix 7.3).

4. Algorithms in MAGIIC

The MAGIIC System Lambert Constant Generation algorithm was analyzed and modeled in Pascal. This algorithm is required for analyzing and modeling the Geographic to Lambert/Polar Grid and Lambert/Polar Grid to Geographic conversions.

The MAGIIC system coordinate conversions were studied, and two interrelated algorithms were analyzed and modeled in Pascal. These were the Polar Grid to UPS and Northing and Easting to UTM conversions. Several inconsistencies have been noted (perhaps only because of the technical writing). Modeling these interrelated algorithms in Pascal led to the decision to use a non-standard Pascal feature - the VAX Pascal external procedure capability. This was considered necessary to best maintain structural integrity, accuracy and clarity in the algorithm representation.

4.1. MAGIIC Lambert Constant Generation Algorithm

The MAGIIC Lambert Constant Generation Algorithm, described in document CG108100A, dated 23 October 1978, paragraph 3.2.119, page 210, has insufficient input parameters and a lack of detail on setting the hemisphere flags to implement the algorithm in Pascal without making several assumptions. If we assume the availability of the underlying spheroid parameters, the Lambert Constant Generation Algorithm performs satisfactorily. The hemisphere flag issue remains unsettled: are they north-south hemispheres, east-west hemispheres, or both? - all three selections make sense in different contexts.

This algorithm has been modeled in Pascal on the VAX computer for uniformity of presentation and for comparison and analysis. All assumptions are included in comments in this Pascal representation (see Appendix 7.3).

4.2. An Interrelated Set of MAGIIC Coordinate Conversion Algorithms

There are a set of four interrelated coordinate conversion algorithms for MAGIIC that warrant a consolidated discussion because although they are pair-wise reciprocal and criss-cross "call" each other, their input and output parameters are specified inconsistently. These algorithms viewed as reciprocal pairs are:

1.	Polar Grid to UPS	(paragraph 3.2.118) reciprocal
2.	UPS to Polar Grid	(paragraph 3.2.117) reciprocal
3.	Northing and Easting to UTM	(paragraph 3.2.86) reciprocal
4.	UTM to Northing and Easting	(paragraph 3.2.85) reciprocal

They can also be viewed as "criss-cross" calling algorithms as follows:

1.	Polar Grid to UPS	(paragraph 3.2.	118) each calls the
			other
2.	Northing and Easting to UTM	(paragraph 3.2.8	6) each calls the
			other
3.	UPS to Polar Grid	(paragraph 3.2.1	17) each calls the
			other
4.	UTM to Northing and Easting	(paragraph 3.2.8	5) each calls the
			other

Analysis of these algorithms and their interrelationships has revealed the following inconsistencies:

- 1. the Polar Grid to UPS algorithm should have an output list consistent with the UPS to Polar Grid algorithms input list since they are reciprocal functions. The lists are not consistent;
- 2. the Northing and Easting to UTM algorithm and the UTM to Northing and Easting algorithm, similarly, have inconsistent output and input lists;
- 3. in both of the above cases the same problems are true when the algorithms are used in the "criss-cross call" sense.

These points will be discussed in greater detail below in the independent analysis of the four algorithms.

4.3. MAGIIC Polar Grid to Universal Polar Sterographic Algorithm

The MAGIIC Polar Grid to Universal Polar Sterographic (UPS) Algorithm, described in document CG108100A, dated 23 October 1978, paragraph 3.2.118, page 209, fails to produce the correct output data.

This algorithm first determines if the UPS or UTM grid reference system is applicable based on the input grid zone designation. The algorithm that handles the UTM conversion is discussed as part of the Northing and Easting to Universal Transverse Mercator conversion algorithm (paragraph 3.2.86 of the previously referenced document).

In an attempt to render this algorithm amenable to analysis and modeling in Pascal several assumptions have been made (based on our interpretation of the composite of various inferences from paragraph 3.2.86, .87, .117, .118). These assumptions are:

- The input/output lists consist of decimal and integer numbers, and alphabetic letters. There are no wholesale conversions of the input/output lists from and to ASCII representation.
- 2. The input/output lists are exchanged with the Northing and Easting to UTM algorithm for processing, when the input grid zone letter is from C thrugh X (not in either polar region). Otherwise, this algorithm performs the processing (for the polar regions).

The following discussion only treats the UPS coverage area where a conversion from PS to UPS should be made. This conversion is handled within the Polar Grid to Universal Polar Sterographic conversion algorithm (paragraph 3.2.118). This algorithm is defective in producing disallowed 100,000 meter squares letter pairs.

The North Polar Area (\geq 84° N) and South Polar Area (> 80° S) are referenced using the UPS grid zone reference system with the grid zone number set to zero followed by one of the grid zone letters A, B, Y, or Z. Letter pairs represent the 100,000-meter squares which overlay the grid zone designations. It is in the lettering of these squares that this algorithm fails.

The descriptions of how to obtain the Easting letter $L_{\rm E}$ from the index $I_{\rm E}$ and the Northing letter $L_{\rm N}$ from the index $I_{\rm N}$ are merely statements that the functions should be performed without any details. Perhaps an elaborated description of the details of these functions would lead to satisfactory conversions, if these details were furnished.

4.4. MAGIIC Northing and Easting to Universal Transverse Mercator Algorithm

The MAGIIC Northing and Easting to Universal Transverse Mercator (UTM) Alorithm as described in document CG18100A, dated 23 October 1978, paragraph 3.2.86, page 156, appears to produce the correct output data on the UTM map segment available for reference. It will certainly fail in grid zones 31V, 32X, 34X, and 36X and should be further evaluated.

This algorithm first determines, whether the UTM or UPS grid reference system is applicable based on the iput grid zone designation. The algorithm that handles the UPS conversion is discussed as part of the Polar Grid to Universal Polar Sterographic conversion algorithm (paragraph 3.2.118 of the previously referenced document).

In an attempt to render the algorithm amenable to analysis and to model in Pascal several assumptions have been made (based on our interpretation of the composite of various inferences from paragraph 3.2.86, .87, .117, .118). These assumptions are:

- 1. The input lists consist of decimal and integer numbers and alphabetic letters. There is no wholesale conversion of the in ut list to ASCII representation. The output lists are entirely in ASCII representation.
- 2. The input/output lists are exchanged with the Polar Grid to UPS algorithm for processing, when the input grid zone letter is not from C through X (for the polar regions). Otherwise, this algorithm performs the processing (for the non-polar regions).

Only the conversion to the UTM coverage area, that is, the actual conversion handled within this Northing and Easting to Universal Transverse Mercator conversion algorithm (paragraph 3.2.86), is discussed here. This

algorithm is defective in producing grid zone designation 31V, which should be truncated at 3° E, and the three non-existent grid zone designations 32X, 34X, and 36X.

5. Comparing the Algorithms Across Systems

5.1. Spheroid Models

Because the Earth is not a perfect sphere, it is modeled as a spheroid. Since this underlying spheroid model of the Earth's oblatness spans most of the coordinate conversion algorithms, the use of various spheroid models was investigated for all four systems. It was found that twelve different spheroid models were used among or in the four systems raising the possibility of inconsistencies that may hamper inter-system communication. Of these spheroid models five were used in common by all four systems. The remaining seven spheroid models were not available in all systems, as illustrated in Table 1. Some of these partially-shared spheroid models may be equivalent, but this cannot be determined until the code is available for all four systems.

5.2. Grid Zone Generation

The Grid Zone Generation algorithms were analyzed for the MAGIIC, Guardrail, Trailblazer, and BETA systems. The Guardrail Grid Zone algorithm text description is consistent with the Trailblazer text description and computer code. The MAGIIC text description differs from the Guardrail and Trailblazer descriptions and would tend to produce less efficient runtime code, but would economize memory. All three of these systems' algorithms would produce the same flawed results, except BETA which fails badly at the upper latitudes. These three versions of the Grid Zone Generation algorithm have been modeled in Pascal.

The BETA Grid Zone Generation algorithm handles details of grid zone generation more completely. The grid zone number calculations handle input longitude beyond -180° and at 180° and beyond whereas the three other systems would fail. The grid zone letter calculations handle the special conditions of grid zone truncation for grid description 31V and the non-existence of grid designations 32X, 34X, and 36X.

Pascal implementations of these algorithms are given in Appendix 7.3. The underlying assumptions are given in the comments included in the code.

5.3. MAGIIC Grid Zone Generation Algorithm

The MAGIIC Grid Zone Generation Algorithm as described in document CG108100A, dated 23 October 1978, paragraph 3.2.90, page 167 has been analyzed and found to handle the following five areas incorrectly:

- 1. the upper limit of longitude $(180^{\circ} E)$,
- 2. the latitudes $\geq 80^{\circ}$ N (considerably below the upper limit, 84° N), where it provides erroneous data,
- 3. the truncated grid zone 31V,
- 4. the non-existent grid zones 32X, 34X, and 36X,
- 5. the regions beyond the stated longitude and latitude limits, where it fails catastrophically.

In general, perhaps due to technical writing, there are many errors of omission and/or commission where the criteria for certain algorithm parts are left unstated; for example, a text states, "If the latitude is 840 north or greater, or 800 south or greater, the grid zone number (sic) shall be set to Y or Z or to A or B, respectively. The grid zone number shall be set to zero."

5.4. Guardrail Grid Zone Generation Algorithm

The Guardrail Grid Zone Generation Algorithm as described in document ESL-TM928, dated 15 Septemer 1979, paragraph 16.6.2.1, page 16-192 fails in five areas:

- 1. at longitudes equal to and beyond 180° E and beyond 180° W.
- 2. at latitudes equal to and beyond 80° N and beyond 80° S,
- 3. at the truncated grid zone 3IV,
- 4. at the non-existent grid zones 32X, 34X, and 36X,

5. beyond the stated longitude and latitude limits, where it fails catastrophicaly.

5.5. Trailblazer Grid Zone Generation Algorithm

The Trailblazer Grid Zone Generation Algorithm is described in the well-commented ROLM assembly language listings. This algorithm, extracted from the code for the GP2UM subprogram dated 20 February 1981, fails in five areas:

- 1. at longitudes equal to and beyond 180° E and beyond 180° N,
- 2. at latitudes equal to and beyond 800 N and beyond 800 A,
- 3. at the truncated grid zone 3IV,
- 4. at the non-existent grid zones 32X, 34X, and 36X,
- 5. beyond the stated longitude and latitude limits.

Because the hierarchical program structure is not yet available for Trailblazer, it is possible that some or all of these problems are handled adequately in higher levels of the program structure.

5.6. BETA Grid Zone Generation Algorithm

The BETA Grid Zone Generation Algorithm, described in document SS22-43, Appendix IV, page II-474 for the ADSONU subprogram, and page II-45D for the ADSCCM subprogram, was in Structured FORTRAN with in-line coding. The "INCLUDE" subprogram ZDBPRO was missing so some "reasonable" assumptions were made about it.

This algorithm performs as specified and effectively handles the following:

- 1. Grid zone wrap-around (longitudes >180° W or \geq 180° E),
- 2. North and South Polar Regions (latitudes \geq 84° N or > 80° S).

- 3. Truncating grid zone 31V,
- 4. The non-existent grid zones 32%, 34%, and 36%.

Table 5-1: Inconsistent Spheroid Usage

(X = spheroid model used in the system)

Spheroid Model		System	L		Technical Manual
	BETA	MAGIIC	GR	TB	TM241-1
Clark 1866	x	x	x	x	x
International	x	X	X	X	X
Clark 1880	X	X	X	X	X
Everest	X	X	X	X	X
Bessel	x	X	X	X	X
Australian	X	X	X		X
Walbeck				X	
Fisher	x			X	
Krasovsky			X	X	
World Geodetic	x		X		
Airy	x				
Malayan	X				
Reference	1	2	3	1 4	

^{1.} DD2642, dtd 20 Feb 81, pg 263

^{2.} CG1808100A, dtd 23 Oct 78, pg 168

^{3.} ESL-TM929, dtd 15 Sep 79, pg 15-158

^{4.} TM32-5811-022-10-0

6. Discussion and Conclusions

6.1. Documentation of Algorithms

Only a design document has been available for the MAGIIC system; and the code has not been available, as shown in Figure 6-1. Therefore, some of the apparent deficiencies in the algorithms may be due to poor technical writing and may not exist in the code itself. For the Guardrail system, tapes containing code have been available, but the printouts obtained to date have been largely unreadable so that suppositions made from the documentation could not be confirmed from the code. In the case of Trailblazer, code is available, but the structured overview expected from documentation is not available.

6.2. Similarity of Functions Across Systems

The functions performed by the geographic transformation algorithms are found to be basically the same across the four systems examined, although the functions are implemented in slightly different ways.

6.3. Incompleteness of Algorithms for Global Applications

The MAGIIC, Guardrail, and Trailblazer transformation algorithms do not account for all the vagaries of the military grid system. Only the BETA algorithms account for all regions and boundaries. The former systems may ensure that "bad" arguments are never passed to these algorithms, so that no anomalies would occur in overall system performance. However, to develop a library of algorithms shared by many systems requires that algorithms internally protect themselves from "bad" input data.

6.4. Robustness of Algorithms

All systems but BETA fail to check for limits of latitude and longitude. This may arise from a common tendency to focus attention on certain areas of the world, e.g. Western Europe. This tendency is especially inappropriate when developing software that may well outlive any given political or geographical constraints.

6.5. Selection/Consolidation of Algorithms

The BETA grid zone generation algorithm is superior to those in the other three systems. Selection of a spheroid model for the library is not possible on the basis of the presently available data and may eventually require developing algorithms based on our experience with many different systems.

Fig. 6-1: Documentation of Algorithms

System	<u>Documentation</u>	Code Available	Comments
MAGIIC	Yes (Barely usable)	No	Bad technical writing: Errors of omission and bad-quality copy. Por- tions of some pages un- readable. TEREC task similar to Guardrail.
Guardrail	Yes	No Glimpses of code appear to be structured FORTRAN; most sections are missing	Multiple Tasks: Program structure "implied" by document's structure. Flowcharts with verbal descrip- tion. At least one significant technique covered by math desc- ription only, entirely included in one box at the flowchart level.
Trailblazer	No Not separately published, but basic documentation included at the beginning of each code segment	Yes assembly	Well commented code. Many similarities to Guardrail.
BETA	Yes	Yes Comprehensive Structured FORTRAN	Program structure explicitely included in documents. Some "key" charts not read- able due to photo reductions. Code is in-line commented from Program Design Language (PDL), but we don't have the PDL

7. Appendices

7.1. Database Entries and Products (PSL/PSA)

The three attached PSA Reports were found to be useful to our analysis. The first report is essentially the PSA Report representing our PSL input data descriptions. The second is the PSA Data Activity Interaction Matrix. It shows the interrelationships between the algorithms and their input (R) and output (D) data items. The third is part of the PSA Structure Chart and is in indented hierarchy chart for the algorithms in our PSL database.

7.2. Algorithm Hierarchy Charts

Structure Report

			STEROHOHT	· · · · · · · · · · · · · · · · · · ·	/
			0.9836 146136		/ I
		10 110111111	IN		
	:	25 AGLPRIFTDE	·	· - · - · · · · ·	1 :
	24	MG_RMSTARGLO	C	·	
	25 10	G_PRECPTFILE -			
	21 MG_LA	MBERTOG		- 1	111
				1 4 4	. 1
	20 HG_MAPIT	ZN	,	1 1	1 1
	19 MG_LDRANCH	·	· /	- i i i i .	
	18 MG_IICOGRDCH	•	1 1 1 1		: : :
	16 MB_IIINUTL	~	/	1 1 1 1 1	1 1
			1 1 1 1 1	1 1 1 1	1 1 .
15	HG_PG2UPS	,	/ 4 1 1 1 1	4 4 4 6	
14 M	9_NEZUTM	/		1 4 1 1 1	
13 NU_ 12 MA ED	AMERCH	/		1 1 1 1 1	1 1 1
12 NG_PR 11 NG_NE2G	HG_PG2UPS G_NE2UTM MUTM2NE AMSRCH	/ 1 1 1 1			1 . :
		1111	ı iriii		i i
				1 1 1	1 1 1
9 MG_GP2L/PG -	/	1 1 1 1 1			1 1 1
7 MG MAPCHE?MF	/ 1 1			1 1 1 1 1	1 1 1
6 MG_IIFLHITZN	/111	1 1111			1 1 1
	1 1 1 1	1 1 1 1 1 1	1 1 1 1 1	1.1.1.1.1.1	1 1 1
5 MG_BUILDSKTCH			1 1 1 1 1 1	1 1 1 1 1	1 1 :
4 MG_GENSPHERE 3 MG_IITARGLOC	/	1 1 1 1 1 1		1 1 1 1 1	1 : !
2 MG_BUILDCD	/	1 1 1 1 1 1		1111	1 1 1
1 MG_Alsorithms				1 1 1 1	1 : :
				11111	1 : i
~	+	~-+	. 4	+	+ - <i></i> -
MG Aldonithms	1 A A I	i A			
MG_Aldorithms	i A Ai i B i	i A	i AAA		! A A
MG_BUILDCD	1 A A! 1 B i 1 B i	1 A 1	I AAA		A A
MG_BUILDCD	I A AI I B I I B I	ì	I AAA	AA A	I AA
MG_Aldorithms	i A A! 1 B i 1 B i 1 B i B B	i A 1 1 1	A A A	A A A	A A
MG_BUILDCD	! B ! B B +	ì	1 AAA	AAA	A A
MG_BUILDCD		 +	- -	A A A	A A A
MG_BUILDCD	1	 +	- -	+ -	A A
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLHITZN MG_MAPCURZNE MG_GP2NF MG_GP2L/PG	1	 	-	+ -	A A A A A A A A A A A A A A A A A A A
MG_BUILDCD	1	 	-	+ -	A A
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLHITZN MG_MAPCURZNE MG_GP2NF MG_GP2L/PG	1	 	-	+ -	A A
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_HAPCUR2NE MG_GP2L/PG MG_GRDZONECNV MG_ME2GP			1	+ -	A A
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN	B B B B B B B B B B		 	+ -	A A A A A A A A A A A A A A A A A A A
MG_BUILDCD MG_IITARGLOC MG_ERNSPHERE	B B B B B B B B B B			+ -	A A
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN	B B B B B B B B B B		 	+ -	1 A A
MG_BUILDCD MG_IITARGLOC MG_IITARGLOC MG_BUSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCUR2NE MG_GP2NE MG_GPZNE MG_GRDZONECNV MG_NE20P MG_FRAMSRCH MG_FRAMSRCH MG_NE2UTH MG_NE3UTH	B B B B B B B B B B		; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	+ -	A A
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCUR2NE MG_GP2L/PG MG_GP2L/PG MG_GRDZONECNV MG_NE2GP MG_FRAMSRCH MG_UTM2NE MG_NE2UTM MG_NEZUTM MG_REZUTM MG_REZUTM MG_RIINUTL MG_RMSINUTL					6 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1
MG_BUILDCD MG_IITARGLOC MG_IITARGLOC MG_BENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCUR2NE MG_GP2NE MG_GPZNE MG_GRDZONECNV MG_NE2GP MG_FRAMSRCH MG_PRAMSRCH MG_PZUPS MG_RBZINVTL MG_RBSINVTL MG_RIICOORDCNV	B B B B B B B B B B B			+	
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCUR2NE MG_GP2L/PG MG_GP2L/PG MG_GRDZONECNV MG_NE2GP MG_FRAMSRCH MG_UTM2NE MG_NE2UTM MG_NEZUTM MG_REZUTM MG_REZUTM MG_RIINUTL MG_RMSINUTL	B B B B B B B B B B B		1		6 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
MG_BUILDCD MG_IITARGLOC HG_GENSPHERE MG_BUILDSKTCH MG_IIFLHITZN HG_MAPCURZNE HG_GP2NE HG_GP2L/PG HG_GRDZONECNV MG_NE2GP HG_FRAMSRCH HG_NE2UTH HG_PG2UPS HG_FRAMSNUTL HG_RMSINUTL HG_IICODRDCNV HG_HAPITZN	B B B B B B B B B B B		1	+	
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCUR2NE MG_GP2NE MG_GP2NE MG_GRDZONECNU MG_NE2GP MG_FRAMSRCH MG_UTMZNE MG_PG2UPS MG_PG2UPS MG_IIINVTL MG_NE2UPM MG_IIINVTL MG_IIIOORDCNU MG_LORANCNU MG_LORANCNU MG_LORANCNU MG_MAPITZN MG_LAMBERTCG	B B B B B B B B B B B		1		A A A
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCUR2NE MG_GP2NE MG_GP2NE MG_GPZVPG MG_GRDZONECNV MG_NE2GP MG_FRAMSRCH MG_UTM2NE MG_NE2UTH MG_PG2UPS MG_IIINUTL MG_RESINUTL MG_IICOORDCNV MG_HAPITZN MG_HAPITZN MG_HAPITZN MG_HAPITZN MG_LAMBERTCG NG_NAWCORN			1	B	
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCURZNE MG_GP2NP MG_GP2NP MG_GRDZONECNV MG_NE20P MG_FRAMSRCH MG_UTMZNE MG_PG2UPS MG_IIINUTL MG_RRSINUTL MG_IIOORDCNV MG_LORANCNV MG_LORANCNV MG_LORANCNV MG_LAMBERTCG MG_NG_STANGLOCON MG_LAMBERTCG MG_NG_NG_MG_NAVCORN	B B B B B B B B B B B			; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	; ; ! ! !
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCURZNE MG_GPZNE MG_GPZNE MG_GPZNE MG_GRDZONECNV MG_NE2GP MG_FRAMSRCH MG_UTM2NE MG_NE2UTH MG_PGZUPS MG_IIINUTL MG_RGUPS MG_ICOORDCNV MG_LAMBERTCG MG_NAVCORN	B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B			B E	; ; ; ; ;
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLHITZN MG_MAPCURZNE MG_GP2NE MG_GP2NE MG_GRDZONECNV MG_NEZOP MG_FRAMSRCH MG_UTMZNE MG_PGZUPS MG_FRANSNCH MG_PGZUPS MG_IIINUTL MG_RGSIINUTL MG_RASINUTL MG_HAPITZN MG_LAMBERTCG MG_MAVCORN MG_PRECPTFLE MG_RRSTARGLOC MG_PRIPTDET	B B B B B B B B B B B B B B		1	B E	; ; ! ! !
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCURZNE MG_GPZNE MG_GPZNE MG_GRDZONECNV MG_NE2GP MG_FRAMSRCH MG_UTMZNE MG_PGZUPS MG_IIINVTL MG_PGZUPS MG_IIINVTL MG_HCORDCNV MG_HCORNCNV MG_HAPITZN MG_LAMBERTCG MG_MAVCORN MG_PREYFFLE MG_NEZMAPCUR MG_NEZMAPCUR	B B B B B B B B B B B			B E	, , , , , , , , , , , , , , , , , , ,
MG_BUILDCD MG_IITARGLOC HG_GENSPHERE HG_BUILDSKTCH MG_IIFLHITZN HG_MAPCURZNE HG_GPZL/PG HG_GPZL/PG HG_GRDZONECNV MG_NE2GP HG_FRAMSRCH HG_UTTXNE HG_NE2UTH HG_PGZUPS			; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	B	B
MG_BUILDCD MG_IITARGLOC MG_GENSPHERE MG_BUILDSKTCH MG_IIFLMITZN MG_MAPCURZNE MG_GPZNE MG_GPZNE MG_GRDZONECNV MG_NE2GP MG_FRAMSRCH MG_UTMZNE MG_PGZUPS MG_IIINVTL MG_PGZUPS MG_IIINVTL MG_HCORDCNV MG_HCORNCNV MG_HAPITZN MG_LAMBERTCG MG_MAVCORN MG_PREYFFLE MG_NEZMAPCUR MG_NEZMAPCUR	B B B B B B B B B B B			B B	B

SEMANTIC-MAP-TEST

Utilizes Matrix

		23 24 Mi 23 Mg_I 22 Mg_RMi 21 Mg_IIIN		ARGLOS	/ / :	- / I I I I I
1 A M	18 MG_ 17 MG_NE; 16 MG_GP2NI MG_HAPCUR2I G_IIFLMITZN GENSPHERE TARGLOC OHGHT	MG_UTM2NE G_BP2L/PG GRDZONECNU 2GP	/			
8 MG_PRECPTFLE - 7 MG_MAVEDRN 6 MG_MAPITZN 5 MG_LORANCNV 4 MG_IICOORDCNV 3 MG_FRAMSRCH 2 MG_BUILDSKTCH 1 MG_BUILDCD	/ / / / 					1 i i i i i i i i i i i i i i i i i i i
1 MG_Algorithms	! !	S S S S S S S S S S S S S S S S S S S			i 	; ; ; ;
6 MG_GP2ME 7 MG_ME2GP 9 MG_FRAMSRCH 9 MG_UTH2NE 0 MG_ME2UTH	1 1 1	1 1 1	U 1	ט ט ט	! ! U U ! U U	1 1 1
11 MG_P02UPS	; ; ;	1	ו ט	ט טטו	1 U 1 U 1 U	
16 MG_NAVCORN	i ·	1	ט ט ט ז וייט ט ט ז		1 U U U 	1 U U U
21 MG_STEROHGHT	1	 	u u		1 U +	 -

Data Activity Interaction Matrix

32 MG_STEROHGHT 31 MG_RMSTARGLOC		, ,
30 MG_RMSINVTL 29 MG_RMSFLMITZN	•	1 1 1 1
28 MG_RMSCOORDENV	, , ,	1 1 1 1
27 MG_RAD2DEG		1 1 1
26 MG_PROXSRCH	- /	1 1 1
1 MG_Angle_in_Degrees/Mins/Secs	l D	1
2 MG_Angle_in_Scaled_Pi_Radians	I R	
3 MG_P75T2	1	1
4 MG_Selieroid	1	I DRI
5 MG_F171T5	1	1
	ţ	
6 MG_Latitude/Longitude	i	i i
7 MG_Northing/Easting 8 MG_Grid_Zone_No./Letter	1	l DFI (DRI
7 MG_xy_Map_Cursor_Coordinate	!	ו נו או
10 MG_UTM_Northing/Easting_Set	, 1	, . 1 1
TO HOLD HILKO, WITH 3, E89 OTH 3 LOE V	+	++
11 MG_Grid_Zone_Letter	1	1 7 1
12 MG_N/E_Letters	1	I R I
13 MG_N/E_Numbers	1	I R I
14 MG_100K_Meter_Square_ID	1	l Ri
	· • · · · · · · · · · · · · · · · · · ·	÷

		25 HG_FRIFIDET	-	1 1 1 1 1 1		/	
	*	24 NG PRECEIFLE	- I				
	S. C.C.	22 HG NESULM				 	
	21 MG_	21 MG_NEZHAFCUR	:	:	 		
	20 NG_ME2GF	3F	:	:	 	 	
	19 MG_NAVCORN	III		, , , , ,	<u>-</u> -	<u>-</u> -	
	17 MG_MAFCURZNE				 	- - 	
16	16 MG_LORANCNU		`	 	 	- -	
DE 51	15 MG_LAMBERTCG	!	· -	- - -	- -	- - -	
14 MG_L/FG2GF	14 MG_L/F626F		 		 		
12 MG_IIINVIL	TL	1 /	 	- <u>-</u> - <u>-</u>	 	 	
11 NG_HFLWITZW		1 1 /	 	 	 	 	
10 MG_IICOORDENU			. – . . – .	. – . . – .	. - .	· _ ·	
CONECNO	/	 	 	 	 	 	
7 MG_GF2L/FG	/	 	 	 	- -	 	
	/	 	 	 	 		
4 MG_DEG2KAD	1 1 1 /	 	 	 	. 		
2 MG_BUILDCD		 	 	 	 	 	
				- !	- !		
G_Ansle_in_Desrees/	w s						
	-	_	R RI	ند		-	
4 KG_Stherold	 :::::::::::::::::::::::::::::::::::		 		iz iz	 œ	
6 M6_Latitude/Lonsitude	. —	¥		1	ين	·	
8 46 Grid, Zone No. /Letter		 =	 ≃ن≃		X (7 X (7 X		
7 MG_ROTHER_Cursor_Coordinate 1 10 MG_UTH_Northing/Easting Set 1				ಚ	- - 		
	· · · · · · · · · · · · · · · · · · ·	÷ -	÷ -	•	:		
MG N/E Letters	- -	- -			- -	 - =	
13 NG.HZE Humbers						 -	
	· <u>-</u>	• ÷	· -			-	

```
1 MG_Alsorithms
    2
        MG_BUILDCD
    3
           MG_IITARGLOC
              MG_GENSPHERE
    2
        MG_BUILDSKTCH
     3
           MG_IITARGLOC
              MG_GENSPHERE
     3
           MG_IIFLMITZN
     3
           MG_MAPCUR2NE
              MG_GP2NE
                 MG_GENSPHERE
       5
                  MG_GP2L/PG
       5
                  MG_GRDZONECNV
      4
              MG_GRDZONECNV
      4
              MG_NE2GP
       5
                  MG_GF2NE
                     MG_GENSFHERE
        6
                     MG_GF2L/PG
        6
                     MG_GRDZONECHV
    2
        MG_FRAMSRCH
     3
           MG_NE2GP
              MG_GP2NE
       5
                  MG_GENSPHERE
                  MG_GF2L/PG
       5
                  MG_GRDZONECNV
     3
           MG_UTM2NE
      4
              MG_GENSPHERE
              MG_NE2GP
                  MG_GP2NE
                     MG_GENSPHERE
        6
                     MG_GP2L/PG
        ó
                     MG_GRDZONECHY
        6
      4
              MG_NE2UTM
       5
                 MG_PG2UPS
                     MG_NEZUTM
PSA168:Loop detected (see level 4) - Structure truncated.
      4
              MG_FG2UPS
       5
                  MG_NE2UTM
                     MG_FG2UFS
        6
PSA168:Loop detected (see level 4) - Structure truncated.
      4
              MG_UFS2PG
     3
           MG_IIINVTL
      4
              MG_IITARGLOC
                  MG_GENSPHERE
      4
              MG_GENSPHERE
           MG_RMSINUTL
     3
    2
        MG_IICOORDCNV
           MG_GF2NE
     3
      4
              MG_GENSPHERE
      4
               MG_GP2L/PG
      4
              MG_GRDZONECHV
     3
           MG_NE2GP
               MG_GP2NE
      4
                  MG_GENSPHERE
       5
                  MG_GP2L/FG
       5
                  MG_GRDZONECNV
           MG_UTM2NE
     3
               MG_GENSPHERE
               MG_NE2GP
```

```
1 MG_Alsorithms
    2
        MG_BUILDCD
     3
           MG_IITARGLOC
              MG_GENSPHERE
        MG_BUILDSKTCH
    2
     3
           MG_IITARGLOC
              MG_GENSPHERE
     3
           MG_IIFLMITZN
           MG_MAFCUR2NE
     3
              MG_GF2NE
       5
                  MG_GENSPHERE
       5
                  MG_GP2L/PG
       5
                  MG_GRDZONECNV
      4
              MG_GRDZONECNV
      4
              MG_NE2GP
       5
                  MG_GP2NE
                     MG_GENSFHERE
                     MG_GF2L/PG
        6
                     MG_GRDZONECHV
    2
        MG_FRAMSRCH
     3
           MG_NE2GP
              MG_GP2NE
       5
                  MG_GENSPHERE
       3
                  MG_GP2L/PG
       5
                  MG_GRDZONECNV
     3
           MG_UTM2NE
              MG_GENSPHERE
      4
              MG_NE2GP
      4
       5
                  MG_GF2NE
                     MG_GENSPHERE
        6
                     MG_GP2L/PG
        6
                     MG_GRDZONECNV
        6
              MG_NE2UTM
      4
       5
                  MG_PG2UPS
                     MG_NEZUTM
FSA168:Loop detected (see level 4) - Structure truncated.
              MG_FG2UFS
       5
                  MG_NE2UTM
                     MG_PG2UPS
PSA168:Loop detected (see level 4) - Structure truncated.
              MG_UFS2PG
      4
     3
           MG_IIINVTL
              MG_IITARGLOC
      4
                  MG_GENSPHERE
               MG_GENSPHERE
      4
           MG_RMSINUTL
     3
        MG_IICOORDCNV
    2
     3
           MG_GF2NE
               MG_GENSPHERE
      4
               MG_GP2L/PG
               MG_GRDZONECNV
      4
           MG_NE2GP
     3
              MG_GP2NE
                  MG_GENSPHERE
                  MG_GP2L/PG
                  MG_GRDZONECNV
     3
           MG_UTM2NE
               MG_GENSPHERE
               MG_NE2GP
```

```
5
                  MG_GP2NE
        6
                     MG_GENSPHERE
        6
                     MG_GP2L/FG
                     MG_GRDZONECHV
        6
               MG_NEZUTM
                  MG_FG2UFS
                     MG_NE2UTM
PSA168:Loop detected (see level 4) - Structure truncated.
               MG_FG2UPS
                  MG_NE2UTM
                     MG_PG2UPS
PSA168:Loop detected (see level 4) - Structure truncated.
              MG_UPS2PG
     3
           MG_NEZUTM
              MG_FG2UPS
                  MG_NE2UTM
PSA168:Loop detected (see level 3) - Structure truncated.
        MG_LORANCHY
     3
           MG_GENSPHERE
        MG_MAPITZN
     3
           MG_LAMBERTCG
        MG_NAUCORN
     3
           MGLIITARGLOC
              MGLGENSPHERE
     3
           MGLIIFUMITIN
           MG_MAR JRONE
      4
              MGLGFCNE
                 MGLGENSPHERE
                 MG_GF2L/FG
       5
                 MGLGROZONEONV
              MG_GRDZONECNV
      4
              MG_NE2GP
       5
                 MG_GP2NE
                     MG_GENSPHERE
        6
                     MG_GF2L/FG
                     MG_GRDIONEONV
           MG_GF2NE
     3
              MG_GENSFHERE
      4
              MG_GF2L/FG
      4
              MG_GRDZONECNV
           MG_NE2GP
     3
              MG_GF2NE
                 MG_GENSPHERE
                 MG_GP2L/FG
       5
                 MG_GRDZONECHV
           MG_UTM2NE
     3
              MG_GENSPHERE
              MG_NE2GP
       S
                 MGLGFINE
                     MGLGENSPHERE
        6
        6
                     MGLGF2L/FG
        6
                     MG_GROZONECHY
              MG_NE2UTM
       5
                 MG_FG2UFS
                     MG_NEZUTH
PSA168:Loop detected (see level 4) - Structure truncated.
              MG_FG2UFS
       5
                 MG_NEZUTM
                     MG_FG2UFS
```

```
PSA168:Loop detected (see level 4) - Structure truncated.
              MG_UPS2PG
      4
     3
           MG_NE2UTM
      4
              MG_FG2UFS
       5
                  MG_NE2UTM
PSA168:Loop detected (see level 3) - Structure truncated.
        MG_PRECPTFLE
    2
     3
           MG_RMSTARGLOC
        MG_PRIPTDET
    2
           MG_IITARGLOC
     3
              MG_GENSPHERE
     3
           MG_IIFLMITZN
           MG_MAPCUR2NE
     3
              MG_GF2NE
      4
       5
                  MG_GENSPHERE
       5
                  MG_GP2L/FG
       5
                  MG_GRDZONECNV
      4
              MG_GRDZONECNY
      4
              MG_NE2GP
       5
                  MG_GF2NE
                     MG_GENSPHERE
        6
                     MG_GP2L/PG
                     MG_GRDZONECNV
        5
     3
           MG_GF2NE
              MG_GENSPHERE
      4
              MG_GP2L/PG
      4
              MG_GRDZONECNV
     3
           MG_NE2GP
              MG_GP2NE
       5
                  MG_GENSPHERE
       5
                  MG_GP2L/PG
       5
                  MG_GRDZONECNV
     3
           MG_UTM2NE
              MG_GENSPHERE
              MG_NE2GF
       5
                  MG_GP2NE
                     MG_GENSPHERE
        6
        6
                     MG_GP2L/PG
        5
                     MG_GRDZONECNV
              MG_NE2UTM
       5
                  MG_PG2UPS
        6
                     MG_NE2UTH
FSA168:Loop detected (see level 4) - Structure truncated.
              MG_PG2UPS
       5
                  MG_NEZUTM
                     MG_FG2UFS
        6
PSA168:Loop detected (see level 4) - Structure truncated.
              MG_UFS2FG
     3
           MG_NE2UTM
              MG_PG2UPS
      4
                  MG_NE2UTM
PSA168:Loop detected (see level 3) - Structure truncated.
     3
           MG_IIINVTL
      4
              MG_IITARGLOC
       5
                  MG_GENSPHERE
              MG_GENSPHERE
     3
           MG_RMSINVTL
     3
           MG_RMSTARGLOC
           MG_NE2MAPCUR
```

```
MG_GP2NE
       3
                  MG_GENSPHERE
       5
                  MG_GF2L/FG
       5
                  MG_GRDZONECHV
      4
              MG_GP2L/PG
      4
              MG_NE2GP
       5
                  MG_GFICHE
                     MG_GENSPHERE
        6
                     MG_GP2L/PG
                     MG_GRDZONECHV
        MG_FROXSRCH
           MG_MAPCUR2NE
              MG_GF2NE
                  MG_GENSPHERE
       5
                  MG_GP2L/PG
       5
                  MG_GRDZONECHV
      4
              MG_GRDZONECNV
              MG_NE2GP
      4
       5
                  MG_GP2NE
                     MG_GENSPHERE
        6
                     MG_GF2L/FG
                     MG_GRDZONECHV
     3
           MG_GF2NE
              MG_GENSPHERE
      4
              MG_GP2L/PG
      4
              MG_GRDZONECNV
     3
           MG_UTM2NE
              MG_GENSPHERE
              MG_NE2GP
                  MG_GPCNE
       5
                     MG_GENSFHERE
        ó
                     MG GROUVEG
        6
        ó
                     MG_GRTZONECNV
              MG_NE2UTN
       5
                  MG_FG2UFS
                     MG_NEQUTM
PSA168:Loop detected (see level 4) - Structure truncated.
              MG_PG2UPS
       5
                  MG_NEZUTM
                     MG_PG2UPS
PSA168:Loop detected (see level 4) - Structure truncated.
              MG_UPS2PG
    2
        MG_STEROHGHT
     3
           MG_IITARGLQC
      4
              MG_GENSPHERE
     3
           MG_IIFLMITZN
     3
           MG_IIINVTL
      4
              MG_IITARGLOC
       5
                  MGLGENSPHERE
              MG_GENSPHERE
```

4 MG_DEG2RAD

DESCRIPTION:

This alsorithm converts andle in dedrees/minutes/seconds into its equivalent andle in scaled Pi radians.

SOURCES: CG108100A/Part_I

SECURITY: U

RESP PD: JWG

ATTRIBUTE:

VALUE:

Degrees_2_Binars_(Binars)

document 107/01/821

uous

trisonometrs

leaf TBD

abbreviation

tsse_of_source
date_acuurred
Processind_done
mathematical_field

tree_level

requirements_Performance

5 MG_GENSPHERE

FROCESS

DESCRIPTION:

This algorithm determines the appropriate spheroid number corresponding to the input latitude/longitude by scanning a specially constructed map database which contains the relationship between longitude bands/latitude strips and spheroid numbers.

KEYWORDS: alsorithm

SOURCES: 3.2.91

CG108100A/Part_I

SECURITY: U

RESP PD: JWG

ATTRIBUTE:

VALUE:

Seheroid_Generation

document '07/01/82'

none

cartographs

leaf TBD

abbreviation
type_of_source
date_acquired
processing_done
mathematical_field

treellevel requirements performance

7 MG_GP2L/PG FROCESS

DESCRIPTION:

This algorithm converts an input seosraphic coordinate latitude/longitude pair to the equivalent Lambert/polar smid

northing/easting coordinate set.

KEYWORDS: also withm

SOURCES: 3.2.115 CG108100A/Fact_I

SECURITY: U

RESP FD: JWG

ATTRIBUTE: VALUE:

abbreviation Lat/Long_2_Lambert/Folar_Grid

ispelofisource document date_acquired '07/01/82' Processing_done none

mathematical_field cartosraphs

tree_level leaf requirements_Rerformance TBD

8 MG_GF2NE

FROCESS

DESCRIPTION:

This algorithm converts and inputs latitude, longitude pair into its equivalent morthing and easting coordinate set.

KEYWORDS: alsorithm

SOURCES: 3.2.83

CG108100A/Part_I

SECURITY: U

RESP PD: JWG

ATTRIBUTE:

VALUE:

Geographic_2_Northing/Easting

document 107/01/821

none

cartography

leaf TBD

HIINTBOIL:

abbreviation

type_of_source date_acumired

processins_done
mathematical_field

tree_level
requirements_performance

9 MG_GRDZONECNV

FROCESS

DESCRIPTION:

This alsorithm converts input longitude and latitude to UTM/UPS Grid Zone number and letter. It fails at the upper (closed) limits of both longitude and latitude.

KEYWORDS: alsorithm

SOURCES: 3.2.70

CG108100A/Fart_I

SECURITY: U

RESP PD: JWG

ATTRIBUTE: VALUE:

> abbreviation Grid_Zone_Generation ture_of_source document 107/01/821 date_acquired

analyzed/Pascalized Processing_done

mathematical_field cartuaraphy

3 robustness tree_level leaf

requirements_serformance semenulls_satisfactors 'EUWGIDRVGZNG/MGGZDG' references

14 MG_L/PG2GP PROCESS

DESCRIPTION:

This alsorithm converts Lambert/Polar said input coordinates into equivalent seosraphic coordinate latitude/lonsitude.

KEYWORDS: alsorithm

SECURITY: U

RESP PD: JWG

ATTRIBUTE: VALUE:

abbreviation Lambert/Polar_Grid_2_Lat/Long

tspelof_source document
date_acquired '07/01/82'

processing_done none mathematical_field cartographs

tree_level leaf requirements_serformance TBD

15 MG_LAMBERTCG

FROCESS

DESCRIPTION:

This alsorithm calculates the Lambert constants required for use with the Lambert/Polur Grid to UPS and the UPS to Lambert/Polar Grid conversion alsorithms.

KEYWORDS: alsorithm

SOURCES: 3.2.119

CG108100A/Part_I

SECURITY: U

RESP PD: JWG

ATTRIBUTE:

VALUE:

Lambert_Constant_Generation:

document 107/01/321

analszed/Pascalizad

ยอกใ**บร**ักระทั่งร

ó lear

ಜತರು ತಿಂದಿ ಪ್ರತಿಯಾಗಿತ ieuwouloarkou

abbreviation typelofisource date_acquired Processina_done mathematical_field

robustness treellevel

requirements_serformance

references

17 MG_MAPOUR2NE PROCESS

DESCRIPTION:

This alsorithm converts an Ry map cursor position to its equivalent northins/easting or latitude/longitude or sold none number/letter and spheroid output coordinate.

KEYWORDS: alsorithm

SOURCES: 3.2.89 CG108100A/Fart_I

SECURITY: U

RESP FD: JWG

ATTRIBUTE: VALUE:

abbreviation Mar_Cursor_2_Nunthing. Esstina

tamelof_source document date_acquired 107/01/821 processins_done none

mathematical_field cartographs

tree_level middle requirements_serformance TBD

20 MGLNE2GP

DESCRIPTION:

This alsorithm converts and inputs northins/easting set into a latitude/lunsitude Pair.

KEYWORDS: alsorithm

SOURCES: 3.2.84

CG108100A/Part_I

SECURITY: U

RESP PD: JWG

ATTRIBUTE:

VALUE:

Northins/Esstins_2_Lat/Lone

document 107/01/321

none

cartuaraphs

leaf TBD

abbreviation isre_of_source date_acquired Processingldone mathematical_field

tree_level

requirements_performance

31 MG_NE2MAPOUR PROCESS

DESCRIPTION:

This alsorithm converts either northing/easting or latitude/longitude or grid zone number/letter pairs into equivalent xw map pursor position pairs for display.

KEYWORDS: alsorithm

SOURCES: 3.2.88 CG108100A/Part_I

SECURITY: U

RESP PO: JWG

ATTRIBUTE: VALUE:

abbreviation

Northing/Eastins_Duhas_Coreco Uspe_of_source document

107/01/321 date_acuuired Processing_done none

mathematical_field cartusraphs

tree_level middle

requirements_performance

22 MG_NE2UTM PROCESS

DESCRIPTION:

This algorithm converts northing and easting to a

composite UTM pair.

KEYWORDS: alsorithm

SOURCES: 3.2.86 CG108100A/Part_I

SECURITY: U

RESP FD: JWG

ATTRIBUTE: VALUE:

abbreviation Northins/Easting_2_UTM

Usee_of_source document date_soquired 107/01/321

Processing_done none

mathematical_field cartography

tree_level leaf requirements_performance TBD

3 MGLPG2UPS

PROCESS

DESCRIPTION:

This alsorithm converts Folar Grid northins/eastins coordinates into equivalent Universal Folar Sterographic or Universal Transverse Mercator (utilizing the NE2UTM alsorithm).

KEYWORDS: alsorithm

SECURITY: U

RESP PD: JWG

ATTRIBUTE:

VALUE:

abbreviation FolarEGrid_2_UFS typeLof_source document date_acquired 107/01/321

Processing_doneanalyzed/Fascalizedmathematicsl_fieldcarbosrsehs

robustness 3
tree_level leaf

requirements_serformance TBD

references 183**433732878**1

27 MG_RADEDEG PROCESS

DESCRIPTION:

This alsorithm converts an angle in scaled Pi radians into an equivalent angle in degrees/minutes/seconds.

SOURCES: CG108100A/Part_I

SECURITY: U

RESP PD: JWG

ATTRIBUTE:

abbreviation
type_of_source
date_acquired
processins_done
mathematical_field
tree_level

requirements_performance

VALUE:

Radians_2_Degrees

document 107/01/821 none

trisonometrs

leaf TBD

33 MG_UPS2PG

FROCESS

DESCRIPTION:

This alsorithm converts Universal Polar Sterographic into equivalent polar srid coordinates.

KEYWORDS: alsorithm

SOURCES: 3.2.117 CG108100A/Part_I

SECURITY: U

RESP FD: JWG

ATTRIBUTE: VALUE:

abbreviation UF3_2_Polar_Grid

type_of_source document date_acquired 107/01/321

Processing_done none mathematical_field cartography

tree_lavel lasf requirements_performance TBD

34 MG_UTM2NE FROCESS

DESCRIPTION:

This alsorithm converts a UTM coordinate set into the equivalent composite northins and eastins rair.

KEYWORDS: alsorithm

SOURCES: 3.2.85 CG108100A/Part_I

SECURITY: U

RESP PD: JWG

ATTRIBUTE: VALUE:

abbreviation UTM_2_Northins/Eastins

tspelofisource document datelacquired 107/01/821

tree_level leaf requirements_performance TED

7.3. Algorithms in Standard Form

```
100
       PROGRAM DriverLambertConstantGeneration (INFUT:OUTPUT);
 200
 300
       ← DriverLambertConstantGeneration provides an environment for
       C DriverLambertConstantGeneration FROCEDURE
 400
                                                                             }
 500
       Clastic the LambertConstantGeneration Procedure
                                                                            >
 600
                                                                             }
 700
       Interfaces with the LambertConstantGeneration procedure are:
 800
           GLOBAL VARIABLES
                                    Eccentricity
 900
       ₹
                                    SemimadorAxis
1000
       ₹
           INPUT PARAMETERS
                                    Lambdol.
                                                                            }
1100
       ₹
                                    LamindaR
                                                                            >
1200
       ₹
                                    FhiU
                                                                            }
1300
       ₹
                                    FliiL
1400
       ₹
                                    Phi1
1500
       ₹
                                    Phi2
                                                                            ን
1600
       ₹
           OUTPUT PARAMETERS
                                                                            ን
                                    Kerre
1700
                                    Tota
       €
                                                                            }
1800
       ₹
                                    ProJectionComeRadius
                                                                            }
1900
       ₹
                                    LambdaC
                                                                            }
2000
       ₹
                                    SaubredEccentricity
                                                                            ን
2100
       ₹
                                                                            >
                                    Hemisphere
2200
       {
                                                                            }-
2300
       { declarations
                                                                            }
2400
2500
       TYPE
2600
                PiRadians
                                          = REAL;
2700
                ZaroToOne
                                          - REAL;
                Flas
2800
                                          = (northern;southern);
2900
                                                                            3
       VAR
3000
3100
                Eccentricity
                                          : ZeroToOne;
3200
                Semimojurāxis
                                          : REAL;
3300
3400
                Lambdal
                                          : PiRadians;
3500
                LambdaR
                                          : PiRadiansi
3600
                PhiU
                                          : PiRadians;
3700
                PhiL
                                          : PiRadiansi
3800
                Phi1
                                          : FiRadians;
3900
                Phi2
                                          : PiRadians;
4000
                                                                            ን
4100
                Канва
                                          : REAL;
4200
                Iota
                                          : REAL!
4300
                ProjectionComeRadius
                                          : REAL;
                                          : FiRadians;
4400
                LambdaC
4500
                SquaredEccentricity
                                         : ZeroToOne;
4600
                Hemisphere
                                          : Flanii
4700
       €
4800
4900
       PROCEDURE LambertConstantGeneration
5000
                  ( {GLOBAL} Eccentricity
                                                       : ZeroToOne;
5100
                    KGLOBALY SemimadorAxis
                                                       ; REAL!
5200
                    CNI
                               Lambdel
                                                       : FiRadians;
                    (IN)
5300
                               LambdaR
                                                       : FiRadians)
5400
                    CKI
                               PhiU
                                                       : FiRadians;
5500
                    CIN>
                               Phil
                                                       : PiRadiansi
5600
                    KIN
                               Phi 1
                                                       : FiRadians;
5700
                    KINE
                               Phil2
                                                       : PiRadians;
                    COUTS VAR Kares
5800
                                                       : REAL;
                    COUTY VAR Inta
5900
                                                       : REALS
                    COUTY VAR ProjectionConeRadius
6000
                                                      : REAL #
6100
                    KOUTY VAR LambdaC
                                                       : PiRadiansi
                    COUT VAR SquaredEccentricity
6200
                                                       : ZeroToOne;
                    COUTY VAR Hemischere
                                                       : Flad
6300
                                                                 DJEXTERN;
                                      30-1
```

Ì

```
6400
        C Procedure LambertConstantGeneration models the Lambert Constant
 6500
 6600
        4 Generation also with a described in range ash 3.2.119 of document
 6700
        € 06108100s dated 23 October 1978.
 6800
        ₹
 6900
           C J.W.Gillis 4-22-82
                                                                                }
 7000
        ₹
                                                                                }
 7100
           C This procedure ASSUMES that certain data are available as
 7200
           C required by the algorithm but not described as invots in the
 7300
           C reference document. These data are: Eccentricity
 7400
                                                    SemimadorAxis
 7500
        {
 7600
 7700
           C This procedure DOESNOT perform data validation checks that are
 7800
           I not specified in the algorithm description. This is to allow the
 7900
           C aldoritim features to be presented more olearly.
 8000
        €
                                                                                    }
        { executables
 8100
                                                                                    3.
 8200
        ₹
                                                                                    ን
 8300
        REGIN
8400
        ₹
                                                                                    >
 8500
                 { establish GLOBAL variables}
 8400
        €
                                                                                    >
 8700
                 Eccentricity
                                           :-- 0.5;
 8800
                 SemimedonAxis
                                           :- 10.0;
 8700
        ₹
                                                                                    3
 9000
                 C establish INPUT PARAMETERS)
 9100
        ₹
                                                                                    >
 9200
                LambdeL
                                           := 0.25;
 7300
                LambdaR
                                           : - 0.0;
 9400
                PhiU
                                          := 0.125;
 9500
                 PiriL
                                           ; 0.0;
 9600
                Phi1
                                           := 0.0675;
 9700
                 Phi2
                                           :- 0.375;
 9800
                                                                                    >
 9900
        R geho INPUT PARAMETERS
10000
        €
10100
                 WRITELNE
10200
                WRITELN (' setur Eccentricity is 'saccentricity);
10300
                 WRITELN (' setur SemimadorAxis is '/SemimadorAxis);
10400
                 WRITELN (' setur Lambdak is 'rhambdak);
                 WRITELN (' setur LambdaR is ',LambdaR);
10500
                 WRITELN (' setup Phill is ', Phill);
10600
                 WRITELN (' setup Phil is ', Phil);
10700
                 WRITELN (' setur Phil is ', Phil);
10800
10900
                 WRITELN (' setur Phi2 is ',Phi2);
11000
        ₹
                                                                                    ን
11100
                 LambertConstantGeneration ( <60.08AL}
                                                          Eccentricity,
11200
                                                          Semimadorfixis
                                               COLOBAL
11300
                                               (TUD)
                                                          Lowbidge
11400
                                               COUTE
                                                          LambdaR,
11500
                                               COUTS
                                                          PiriU,
                                               COUTS
11600
                                                          Phil.
11700
                                               くりりてき
                                                          Pini1,
11800
                                               {TUO}
                                                          Phil2,
11700
                                               <u >
                                                          Karrisa
12000
                                               {KKI}
                                                          Iulas
12100
                                               <KI}}
                                                          ProjectionConeRadius,
12200
                                               <KIB>
                                                          LambdaCe
12300
                                               {IN}
                                                          SaudredEogentricityr
12400
                                               {KI}
                                                          Hemisthere );
12500
        €
                                                                                    7
                 Clist output renemeters from logy
12600
```

```
12700 {
12800 WRITELN;
12900 WRITELN (' Karra ia ' *Karra);
13000 WRITELN (' Inta is ' *Inta);
13100 WRITELN (' ProjectionConeRadius is '*ProjectionConeRadius);
13200 WRITELN (' LambdaC is '*LambdaC);
13300 WRITELN (' SeparadEccentricity is '*SeparadEccentricity);
13400 WRITELN (' Hemisrhere is '*Hemisrhere)
13500 {
13600 END. { DriverLambertConstantGeneration PROCEDURE
```

Land Chillips

```
100
       MODULE Lederoe (INPUT:OUTPUT);
 200
       ₹
                                                                                     ን
 300
       ₹
 400
       ₹
       TYPE
 500
 600
                  zeroToOne
                                                     = REAL;
 700
                  PiRadians
                                                     - REAL!
 800
                  Flas
                                                     = (nurlinern, southern);
 700
       ₹
                                                                                    3
1000
       €
1100
       €
       PROCEDURE LambertConstantGeneration
1200
1300
                  ( CGLOBAL) Eccentricity
                                                     : ZeroToOne;
1400
                    (GLOBAL)
                               SeminadorAxis
                                                     : REAL ;
1500
                    {NT}
                               Lambdal
                                                     : PiRadiansi
1600
                    {IN}
                              LamindaR
                                                     : FiRadians;
1700
                    {KI}
                               PhiU
                                                     : PiRadians;
1800
                    < NI>
                              Phil.
                                                     : FiRadians;
1900
                    {IN}
                               Phi1
                                                     : FiRadians;
2000
                    {IN}
                               Phi 2
                                                     : PiRadians;
                    (OUT) VAR Karpa
2100
                                                     : REAL;
                    {OUT> VAR Iola
2200
                                                     : REAL;
                    KOUTE VAR ProjectionConeRadius : REAL#
2300
                    COUTS VAR LambdaC
2400
                                                     : FiRadians;
                    KOUTE VAR SagaredEccentricity
2500
                                                    : ZeroToOne;
                    COUTS VAR Hemisshere
2600
                                                     : Flas
                                                                );
2700
2800
       4 Generation alsorithm described in rarastraph 3.2.119 of document.
2900
       € CG108100a dated 23 October 1978.
3000
                                                                                     >
3100
       C J.W.Gillis 4-27-82
3200
       ₹
       C This procedure ASSUMES that certain data are available as
3300
3400
         required by the alsorithm but not described as inputs in the
       ₹
3500
       C reference document. These data are: Eccentricity
3600
                                                Semiimadornais
3700
3800

    This procedure DOESNOT perform data validation checks that are

       C not specified in the alsorithm description. This is to allow the
3900
4000
       { also rithm features to be presented more clearly.
                                                                                     }
4100
       ₹
4200
       €
4300
       €
       CONST
4400
4500
                  PiOver2
                                                     = 1.57079;
4600
       ₹
4700
       VAR
4800
                  Phi
                                                     : ARRAY [1..3] OF Firadians;
                  PhiPrime
4900
                                                     : ARRAY [1..3] OF Firadians;
5000
                                                     : ARRAY [1..3] OF Pikadiansi
                  Zeta
                  Curvalure
                                                     : ARRAY [1..2] OF REAL;
5100
                                                     :INTEGER;
5200
                  Index
5300
       BEGIN
5400
                                                                                     ż
          PhiC13 := Phil;
5500
          PhiC23 :- Phi2;
5600
5700
          PhiC31 34 (PhiU+PhiL)/2.0#
5800
       ₹
                                                                                     >
5900
          LambdaC := (LambdaL + LambdaR)/2.0#
6000
6100
          SquaredEccentricity := SQR(Eccentricity);
6200
          FOR Index :=1 TO 3 DO
6300
                                         30-4
```

```
6400
              PhiPrime[Index] :- ARCTAN((1.0-SeparedEccentricity)*
6500
                                          (SIN(ABS(PhitIndex1))/
6600
                                           COS(ABS(FhiEIndexD))));
6700
                                                                                   >
6800
           FOR Index :- 1 TO 2 DO
6900
              Curvature[Index] :- SemimajorAxis/
7000
                                   (SQRT(1.0-SquaredEccentricits*
7100
                                         SQR(SIN(ABS(Phillindex1))));
           FOR Index :- 1 TO 3 DO
7200
7300
              ZetaCIndex1 :- PiOver2-PhiPrime[Index];
7400
        €
                                                                                   ን
7500
           lota := (LN(COS(ABS(PhiCl1)))-LN(COS(ABS(PhiCl1)))+
7600
                    LN(Curvature[1])~LN(Curvature[2]))/
                   (UN(SIN(ZetaC13/2.0)/COS(ZetaC13/2.0))-
7700
7800
                    LN(SIN(Zeta[23/2.0)/COS(Zeta[23/2.0)));
7900
        €
                                                                                   >
           Kappa := (Curvature[1]*COS(ARS(Phi[1])))/
8000
8100
                    (Inta*((SIN(Zeta[13/2.0)/COS(Zeta[13/2.0))**Inta));
8200
       ₹
                                                                                   }
8300
           ProjectionConeRadius := Kappa*((SIN(Zeta[3]/2.0)/
8400
                                           CDS(Zeta[33/2.0)) ##Iuta);
                                                                                   3.
8500
           K NO ALGORITHM is available for the Hemisphere Flast set >
8600
8700
        €
                                                                                   }
8800
           { list intermediate values }
8900
           WRITELN;
9000
           WRITELN (' computed Phil3] is 'sPhil3]);
9100
           FOR Index := 1 TO 3 DO
9200
              WRITELN (' commuted PhiPrimeEIndex) is 'rPhiPrimeEIndex3);
7300
9400
           FOR Index := 1 TO 2 00
              WRITELN (' computed ConvatureLindex3 is ':ConvatureLindex5);
9500
           FOR Index :- 1 TO 3 DO
9600
              WRITELN (' commuted Zetafindex1 is ',Zetafindex1);
9700
9800
9900
        END; { LambertConstantGeneration PROCEDURE }
10000
10100
        END. { Loseroc MODULE }
```

```
100
                                                                            }
 200
       PROGRAM Drvit (INPUT, OUTFUT);
 300
       ₹
                                                                            ን
 400
                                                                            3.
       ₹
       TYFE
 500
600
              Meters
                                    =REAL;
 700
              Decameters
                                    =REAL;
800
              Letter
                                    ='A'..'Z';
                                    =INTEGER;
 700
               Spheres
                                                                            )-
1000
       VAR
1100
              FGNorthinsCoord
                                    :Meters;
1200
             FGEastingCoord
                                    :Meters;
              PGZoneLetter
1300
                                    :Letter;
             PGZoneNumber
                                    :INTEGER;
1400
              SpheroidNumber
1500
                                    :Spheres:
1600
             UPSZoneLetter
                                    :Letter:
1700
              UFSEastingLetter
                                    :letter:
1800
             UPSNorthingLetter
                                    :Letter;
1900
              UPSEastingNumber
                                    :Decameters;
2000
             UFSNorthinsNumber
                                    :Decameters:
2100
                                                                             >
       ₹
2200
       {
                                                                             }
2300
       PROCEDURE PolarToUPS
2400
                  CCIN 3
                              FGNorthinsCoord
                                                    :Meters;
2500
                   KIN >
                              FGEastingCoord
                                                    :Neters;
2600
                   CIN >
                              F6ZoneLetter
                                                    :Letter;
2700
                   KIN >
                              FGZoneNumber
                                                    :INTEGER;
2800
                   KIN D
                              SaheroidNumber
                                                    :Seheres;
2900
                   KOUTS VAR UPSZoneLetter
                                                    :Letter;
3000
                   (OUT) VAR UPSEastingLetter
                                                    :Letter;
                   {CUT} VAR UPSNorthinsLetter
3100
                                                    :Letter;
3200
                   COUT? VAR UPSEastinsNumber
                                                    :Decameters;
3300
                   COUTS VAR UPSNorthinsNumber
                                                    :Decameters);EXTERN;
3400
       -(
                                                                            >
3500
          BEGIN
                                                                            >
3600
3700
       WRITE! N:
       WRITELN (' ENTER PG NORTHING COORD ');
3800
3900
       READLN (FGNORTHINGCOORD);
4000
       WRITELN;
       WRITELN (' ENTER PG EASTING COORD ');
4100
       READLN (PGEASTINGCOORD);
4200
4300
       WRITELN;
4400
       WRITELN ('ENTER PG ZONE NUMBER ');
4500
       READLN (PGZONENUMBER);
4600
       WRITELN;
4700
       WRITELN ('ENTER PG ZONE LETTER ');
4300
       READLN (FGZONELETTER);
4700
       WRITELMS
5000
       WRITELN (' ENTER SPHEROID NUMBER '); READLN (SPHEROIDNUMBER);
5100
                                                                            >
5200
5300
       FolarToUFS (PGNorthingCoord,
5400
                    PGEastinsCoord,
5500
                    FGZoneLetter:
5600
                    FGZoneNumber,
5700
                    SpheroidNumber,
5800
                    UFSZoneLetter,
5900
                    UFSEastinsLetter,
                    UFSNorthinsLetter,
6000
6100
                    UPSEastinsNumber,
6200
                    UFSNorthingNumber);
6300
                                                                            Э.
       €
```

```
100
                                                                               >
 200
       MODULE PG2UPS (INPUT, OUTPUT);
 300
                                                                               >
       ζ.
 400
       €
 300
       TYPE
                                     =REAL;
 600
               Meters
 200
               Decameters
                                     =REAL;
                                     ='A'..'Z';
 800
               Letter
 900
               Spheres
                                     =INTEGER;
1000
       ₹
                                                                               ን
                                                                               3
1100
       {
1200
1300
       PROCEDURE PolarToUPS
1400
                  \langle \langle \langle \langle \langle \langle \rangle \rangle \rangle \rangle \rangle
                               PGNorthinsCoord
                                                     : Metersi
                    (IN )
1500
                               PGEsstinsCoord
                                                     :Metersi
1600
                    CIN >
                               FGZoneLetter
                                                     :Letier:
                    CIN >
1700
                                                     :INTEGER;
                               FGZoneNumber
                   < NI>
1800
                               SeheroldNumber
                                                     :Seheres;
1900
                    COUTY VAR UPSZoneLetter
                                                     :Letter;
2000
                    COUTS VAR UPSEastingLetter
                                                     :Letter;
                    (OUT) VAR UPSNorthinsLetter
2100
                                                     :Letter;
2200
                    (OUT) VAR UPSEastingNumber
                                                     :Decameters;
2300
                    COUTS VAR UPSNorthingNumber
                                                     :Decameters);
2400
       ۲
2500
2300
       K FROCEDURE FolarToUFS models the Pular Grid to Universal Polar
2700
         Steriographic conversion algorithm described in Paragraph
2800
       3.2.118 of CG108100A dated 23 October 1978.
2700
3000
       C PROCEDURE PolarToUPS CALLs: PROCEDURE ConventToUPS
3100
                                         PROCEDURE ConvertToUTM
3200
3300
       { Programmed by J.w.sillis
                                           5-5-82
3400
3500
       { This procedure ASSUMES that certain data are available as
3600
       K required by the algorithm but not adequately described in the
3700
       { slsorithm description.
3800
3900
       { This procedure DOESNOT perform data validity checks that are
4000
       4 not specified in the algorithm description. This is to allow
4100
       3 the algorithm features to be presented more clearly.
4200
       C PROCEDURE FolarToUFS accepts northing and easting coordinates.>
4300
4400
       {\mathfrak C} tests whether the UPS or UTM grid system is a suitable target {\mathfrak F}
       { system. If the UPS grid system is appropriate, the the
4500
       Conversion is performed by this procedure. If the UTM grid
4500
4700

    System is Speropriate, then this procedure CALLs the

4300
       4 FolseToUTM procedure (slaorithm 3.2.86) to perform the
4900
       € conversion.
5000
       £
                                                                               ን
3100
       ₹
5200
       TYPE
5300
3400
              Meters
                                    = REAL ;
5500
              Decameters
                                    = REAL;
5600
              Letter
                                    = 'A',,'Z';
5700
              Spheres
                                    = INTEGER;
5800
                                                                               3
9700
       VAR
5000
              UTMZoneLetter
                                    :Letter;
3100
              UTMEastingLetter
                                    :Letter;
6200
              UTMNorthinsLetter
                                    :Letter;
6300
              UTMEastingNumber
                                     :Decameters;
```

```
5400
              UTMNorthingNumber
                                   :Decameters;
                                                                            3
 6500
 3300
        PROCEDURE ConvertToUPS
 a700
                   (KIN )
                              FGNorthinsCoord
                                                   :Meters;
 a800
                    CIN >
                              FGEastingCoord
                                                   :Meters;
 5700
                    CIN >
                              FGZoneLetter
                                                   :letter;
 7000
                              FGZoneNumber
                                                   :INTEGER;
                    C MID
 7100
                    (OUT) VAR UPSZoneLetter
                                                   :Letter;
 7200
                    (OUT) VAR UPSEastingLetter
                                                   :Letter;
 7300
                    (OUT) VAR UPSNorthingLetter
                                                  :Letter;
 7400
                    (OUT) VAR UFSEastingNumber
                                                   :Decameters;
 7500
                    COUTS VAR UPSNorthingNumber :Decameters);EXTERN;
 7600
                                                                            }
 7700
        PROCEDURE ConvertToUTM
 7300
                              FGNorthinsCoord
                                                   :Meters;
                   (KIN )
 7900
                    CIN >
                              PGEastinsCoord
                                                   :Meters;
 3000
                    C MI>
                              FGZoneLetter
                                                   :Letter;
                    KIN >
                                                   : INTEGER;
 8100
                              19danuNeno589
 3200
                    CIN >
                              SpheroidNumber
                                                   :Seheres;
 8300
                    KOUTE VAR UTMZoneLetter
                                                   :Letter;
                    COUT? VAR UTMEastingLetter
 8400
                                                   :Letter;
                    COUTS VAR UTMNorthinsLetter
                                                  Letteri
 3500
 8600
                    COUTY VAR UTMEastingNumber
                                                   :Decameters;
 3700
                    (OUT) VAR UTHNorthinsNumber
                                                  :Decameters);EXTERN;
                                                                            >
 8800
        ί.
 3700
                                                                            Э.
        ₹
           (Select the appropriate grid reference system
 7000
 9100
 9200
           BEGIN
 9300
        WRITELN ( ' PG2UPS ALL HOOKED UP');
 9400
              IF PGZoneNumber = 0
 9500
                  THEN ConvertToUPS ({OUT} FGNorthinsCoord,
                                      (OUT) FGEastinsCourd:
 9300
 9700
                                      {OUT} FGZoneLetter,
                                      COUTS PGZoneNumber,
 9800
                                      {IN } UFSZoneLetter,
 7900
                                      (IN > UPSEastinsLetter,
10000
                                      KIN > UPSNorthingLetter,
10100
                                      {IN } UFSEastingNumber,
10200
                                      CIN > UPSNorthingNumber)
10300
                  ELSE ConvertToUTM ({OUT} PGNorthinsCoord,
10400
                                      (OUT) PGEastinaCoord,
10500
                                      (OUT) PGZoneLetier,
10600
                                      (OUT) FGZoneNumber,
10700
                                      {OUT} SpheroidNumber,
10300
                                      (IN ) UTMZoneLetter,
10900
11000
                                      (IN ) UTMEastinsLetter,
                                      KIN > UTMNorthinsLetter,
11100
                                      (IN ) UTMEastinsNumber,
11200
                                      (IN ) UTMNorthinsNumber);
11300
11400
                                                                            >
11500
        END; (of PROCEDURE PolarToUPS)
11600
                                                                            >
11700
11300
        END. (of MODULE PG2UPS)
11700
```

```
100
       ₹
                                                                          ን
 200
       MODULE CNV2UFS (INPUT, OUTPUT);
 300
                                                                          >
 400
       ۲
 500
       TYPE
 300
              Meters
                                   ≕REAL;
 700
              Decameters
                                   =REAL;
                                   ='A'..'Z';
 600
              Letter
900
       ₹
                                                                          ን
       FROCEDURE ConvertToUPS
1000
                 (KIN >
1100
                          FGNorthinsCoord
                                                  :Meters;
1200
                  < NI>
                             FGEastingCoord
                                                  :Meters;
1300
                  (IN }
                             PGZoneLetter
                                                  :Letter;
1400
                  CIN >
                             FGZoneNumber
                                                  : INTEGER;
                  (OUT) VAR UPSZoneLetter
1500
                                                  :Letter;
1600
                  (OUT) VAR UFSEastingLetter
                                                  :Letter;
1700
                  (OUT) VAR UPSNorthingLetter
                                                  :Letter;
1300
                  (OUT) VAR UPSEastinsNumber
                                                  :Decameters;
1900
                  (OUT) VAR UPSNorthinsNumber
                                                  :Decameters);
2000
2100
       ₹
2200
       -C PROCEDURE ConvertToUFS performs the Polar Grid to Universal
2300
       -C Polar Steriographic conversion algorithm described in
2400
       \prec paragraph 3.2.118 of CG108100A dated 23 October 1978.
3500
2600

    PROCEDURE ConvertToUPS is CALLed by: PROCEDURE PularToUPS

2700
                                                PROCEDURE PolarTouTM
2800
2700
       { Frosrammed by j.w.gillis
                                         5-6-82
3000
       4 This procedure ASSUMES that certain data are available as
3100
3200
       { required by the algorithm but not adequately described in the
3300
       { also withm description.
3400
3500
       6 This procedure DOESNOT perform data validity checks that are
3600
       { not specified in the algorithm description. This is to allow
3700
       If the alsorithm features to be presentes more clearly.
3800
3900
       4 FROCEDURE ConvertToUPS accepts polar grid northing and easting.
4000
       { coordinates and converts them to UFS coordinates.
                                                                          þ
4100
                                                                          ŀ
4200
       4
                                                                          }-
       TYPE
4300
4400
                                  = REAL;
             Meters
4500
                                  = REAL;
             Decameters
4500
             Letter
                                  = 'A'..'Z';
4700
       €.
                                                                          }
4800
       VAR
1200
             Index
                                  : INTEGER;
5000
                                  : ARRAY [1..26] OF Letter
             GridLetter
5100
       FUNCTION aMODD ({IN} a,b:REAL):INTEGER;
5200
5300
                                                                          }
5400
          VAR Ainteser
                                  : INTEGER;
                                  : INTEGER;
5500
              Binteser
5600
                                                                          }
5700
          BEGIN
2800
             Ainteser: = TRUNC(a);
5700
             Binteser:=TRUNC(b);
3000
             aMODb:=Ainteser MOD Binteser
       END; (of FUNCTION MODab)
6100
6200
       ₹
       {
6300
                                 30-10
```

```
5400
        0
                                                                            ን
 6500
           BEGIN
 6600
           ť
 3700
5800
              (Initialize the GridLetter array.
6900
                                                                            ን
              FOR Index := 1 TO 26 DO
7000
 7100
                  GridLetterCIndex3:=CHR(ORD('A')+Index-1);
7200
7300
              ₹
 7400
              ₹
 7500
              (Calculate the UPSEastingNumber & UPSEastingLetter Index
 7600
 7700
              IF PGZuneLetter > 'M'
                 THEN BEGIN
7800
 7900
                          UFSEastingNumber: =aMODb((FGEastingCoord-200000),
8000
                                                    100000)/10;
8100
                          Index:=aMODb((FGEastingCoord-200000)/1000000,20)
8200
              END; Cuf IF>
8300
                                                                            ን
              IF PGZoneLetter < 'N'
3400
                 THEN BEGIN
8500
8600
                          UFSEastingNumber: =aMODb(FGEastingCoord,
                                                    100000)/10;
8700
                          Index:=aMODb(FGEastinsCoord/100000,18)
8800
 8900
              END; COT IFD
                                                                            >
 9000
        WRITELNS
 7100
        WRITELN ('UPSEASTINGNUMBER IS ',UPSEASTINGNUMBER);
 9200
        WRITELN ('AND ITS LETTER INDEX IS ', INDEX);
 7300
 7400
              (Calculate the UPSEastinsLetter from its index.
 9500
9600
              IF Index > 16
9700
                 THEN Index:=Index+2;
7300
              IF (Index > 2) AND (Index < 17)
9900
10000
                  THEN Index:=Index+1;
10100
              UFSEastingLetter:=GridLetter[Index];
10200
              ٤
10300
        WRITELNE
        WRITELN ('UPSEASTINGLETTER IS ', UFSEASTINGLETTER);
10400
        WRITELN ('AND ITS CORRECTED INDEX IS ', INDEX);
10500
10600
10700
               {
              «Calculate the UFSNorthingNumber & NorthingLetter Index
10800
10900
11000
              IF FGZoneLetter > 'M'
11100
                  THEN BEGIN
                          UPSNorthingNumber:=aMODb((FGNorthingCoord-1300000).
11200
11300
                                                     100000)/10;
                          Index:=aMODb(((FGNorthinsCoord-1300000)/100000);
11400
                                        24)
11500
              END; (of IF)
11600
11700
                                                                            }
              IF PGZoneLetter < 'N'
11800
11700
                  THEN BEGIN
                          UPSNorthingNumber:=aMODb((PGNorthingCoord-800000).
12000
                                                     100000)/10;
12100
12200
                          Index: =aMODb(((FGNorthinsCoord-800000)/100000);
12300
                                        24)
12400
              END; (of IF)
12500
12600
                                      30-11
```

```
12700
        WRITELN;
        WRITELN ('UPSNORTHINGNUMBER IS ', UPSNORTHINGNUMBER);
12800
12900
        WRITELN ('AND ITS LETTER INDEX IS ', INDEX);
              (Calculate the UPSNorthinsLetter from its Index
                                                                           >
13000
13100
13200
              IF Index > 16
                 THEN Index:=Index+2;
13300
              IF Index ≤ 17
13400
                 THEN INDEX:=Index+1;
13500
              UPSNorthingLetter:=GridLetter[Index];
13600
13700
13800
        WRITELNS
        WRITELN ('UPSNORTHINGLETTER IS ', UPSNORTHINGLETTER);
13700
14000
        WRITELN ('AND ITS CORRECTED INDEX IS ', INDEX);
14100
              €
14200
              ₹
                                                                           }
14300
14400
        -{
        END; (of PROCEDURE ConvertToUFS)
14500
                                                                           }
14600
        END. (of MODULE CNV2UFS)
14700
```

TEMPORARY

STUB PROCEDURE

```
100
 200
       MODULE CNV2UTM (INPUT, OUTPUT);
 300
 400
       TYPE
 500
 600
               Meters
                                    =REAL;
 700
                                    =REAL;
               Decameters
 800
                                    =SET DF CHAR;
              Letter
 900
               Spireres
                                    =INTEGER;
1000
       €
                                                                            >
1100
1200
       PROCEDURE ConvertToUTM
1300
                  ({IN }
                             FGNorthinsCoord
                                                   :Meters:
1400
                   (IN)
                             F'GEastinsCoord
                                                   :Matacsi
1500
                   CIN >
                             FGZoneLetter
                                                   :Letter;
                             FGZoneNumber
1600
                   \{IN\}
                                                   : INTEGER;
1700
                   <IN >
                              SpheroidNumber
                                                   :Seheresi
1800
                   {DUT} VAR UTMZoneLetter
                                                   :Letter:
1900
                   {OUT} VAR UTMEastinsLetter
                                                   : | etter:
2000
                   {OUT} VAR UTMNorthingLetter
                                                   :Letter;
                   {OUT} VAR UTMEastingNumber
2100
                                                   :Decameters;
2200
                   (OUT) VAR UTMNorthinsNumber
                                                   :Decameters);
2300
       ₹
2400
2500

    FROCEDURE ConvertToUTM performs the Polur Grid to Universal

2600

    Transverse Mercator conversion alsorithm described in

       f paragraph 3.2.86 of CG108100A dated 23 October 1978.
2700
2800
2900

    PROCEDURE ConvertToUTM is CALLed by: PROCEDURE PolarToUTM

3000
       ₹
                                                 FROCEDURE FolarToUFS
3100
3200
       { Programmed by j.w.gillis
                                         5-6-82
3300
3400
       { This procedure ASSUMES that certain data are available as
3500
       { required by the alsorithm but not adequately described in the
3600
       { alsorithm description.
3700
       { This procedure DOESNOT perform data validity checks that are
3800
       { not specified in the alsorithm description. This is to allow
3900
4000
       f the alsorithm features to be presentes more clearly.
4100
4200
       { PROCEDURE ConvertToUTM accepts polar grid northing and easting}
4250
       { coordinates and converts them to UTM coordinates.
4900
       ₹
5000
       TYPE
5100
5200
             Meters
                                   = REAL;
5300
              Decameters
                                   = REAL;
                                   = SET OF CHAR;
5400
             Letter
                                   = INTEGER;
5500
              Spileres
                                                                            >
5600
       {
5700
       ₹
          BEGIN
5800
5900
             IF
                {THEN}
6000
       WRITELN (' CNV2UTM HOOKED UP OK')
6002
6100
6200
       END; {of PROCEDURE ConvertToUTM}
6300
6400
       END. (of MODULE Crivauth)
3500
                                      30-13
```

```
100
 200
        FROGRAM ConvertToUTM (INFUT, OUTFUT);
 300
        \{ \}
 400
        TYPE
 500
          ASCIIArras
                             = ARRAY [1..4] OF CHAR;
 500
                             = CHAR;
           Letter
 700
       43
 800
       VAR
 700
           UTMCoordinate
                                 : RECORD
1000
              UTNZoneNumber
                                 :ARRAY[1..2] OF INTEGER;
              UTMZoneLetter
1100
                                 :Letter;
1200
              UTMEastinsLetter :Letter;
1300
              UTMNorthingLetter :Letter;
              UTMEastingNumber :ARRAY[1..4] OF CHAR;
1400
              UTMNorthinsNumber :ARRAYC1..43 OF CHAR
1500
           END; (UTMC )
1600
1700
       \langle \cdot \rangle
1800
           PGNorthinsCoord
                              :INTEGER; <IN>
1900
           PGEastingCoord
                              :INTEGER; {IN>
           FGZoneLetter
2000
                              :Letter ; {IN}
2100
           PGZoneNumber
                               :INTEGER; (IN)
2200
           SpheroidNumber
                              :INTEGER; (IN)
2300
           I, EletterOffset, Tel, Tp, Tn1: INTEGER; (Local)
2400
                               : INTEGER;
           Mumber
2500
       €):
2600
       ()
2700
       K PROCEDURE ConvertToUTM performs the Pular Grid to Universal
2800
       4 Transverse Mercator conversion algorithm described in
       { saradraph 3.2.86 of CG108100A dated 23 October 1978.
2900
3000
       \cdot
3100

    FROCEDURE ConvertToUTM is CALLed by: FROCEDURE FolarToUTM

3200
                                                 PROCEDURE PolarTours
3300
       {}
3400
       4.3-
3500

« This procedure ASSUMES that certain data are available as

3600
       4 required by the alsorithm but not adequately described in the
3700
       { alsorithm description.
3800
3900
       4 This procedure DOESNOT perform data validity checks that are
1.0
       { not specified in the alsorithm description. This is to allow
4100
       4 the algorithm features to be presentes more clearly.
4200
4300
       ( PROCEDURE ConvertToUTM accepts polar grid northing and easting)
4400
       Coordinates and converts them to UTM coordinates.
4500
       {}
4600
4700
         PROCEDURE ConvertToASCII( Number: INTEGER (in);
4800
                                    VAR AlphaNum: ASCIIArras ) #
4900
       {}
           K Convert an integer number to its ASCII representation in base 10 0
5000
5100
            { This procedure not detailed in source }
5200
       3.7
                 I: INTEGER;
5300
                AdJ: INTEGER;
5400
5500
       <>
5300
           BEGIN
5700
             FOR I := 4 DOWNTO 1 DO
2800
                  REGIN
5700
                    AdJ := (Number MOD 10) + GRD('0') ;
                    AlphaNumEII := CHR(AdJ) ;
5000
5100
                     Number := Number DIV 10
6200
                  END
3300
       -00
```

```
3400
         END; (ConvertToASCII)
 5500
        \langle \rangle
 3400
        ( )
 6700
        BEGIN (ConvertToUTM)
 4800
        47
 6200
        { read insut }
 7000
        ()
 7100
        WRITELME
        WRITELN ('PG Northing Coord (integer) ') ;
 7200
 7300
        READLN (PGNorthinsCoord) ;
 7400
        WRITELN ('PG Eastins Coord (inteser) ') ;
 7500
        READLN (PGEastingCoord) ;
 7600
        WRITELN ('FG Zone Letter (Letter) ') ;
 7700
        READLN (PGZoneLetter) ;
7800
        WRITELN ('PG Zone Number (integer) '') ;
 7900
        READLN (FGZoneNumber) ;
 8000
        WRITELN ('Ssheroid Number (integer) ') ;
 8100
        READLN (ScheroldNumber) ;
 3200
 3300
        € end of insut }
 3400
        -(\cdot)
 8500
           WITH UTMCoordinate DO
 8400
              BEGIN
 3700
            { Find Easting Letter (A2) }
 3800
             I := PGZuneNumber MOD 3 ;
 8900
            { Easting Letter Offset Factor }
 9000
            CASE I OF
 9100
               O: ELetterOffset := 24;
 9200
               1: EletterOffset := 6;
 9300
               2: ELetterOffset := 15
 9400
            FNn:
 9500
        <>>
 9600
            Tel := ( PGEastingCoord - 500000 ) BIV 100000 + EletterOffset;
 9700
        \{\}
 9800
        4 Following letter conversion not according to documentation; >
 9900

    Procedure in documentation fails at least at Bad Hersfeld, FRG,

        4 which is in 32U, and this one succeeds at least there.
10000
10100
        ()
            IF Tel > 15 THEN Tel := Tel + 2
10200
10300
               ELSE IF Tel > 10 THEN Tel := Tel - 1
10400
               ELSE Tel := Tel - 2 ;
             Tel :- Tel + ORB('A') - 1 ;
10500
10600
            UTMEsstinsLetter := Chr( Tel ) (A2);
10700
10800
            3 Find Northins Number 3
10900
        IF NOT ODD( PGZoneNumber) THEN
11000
           F6NorthingCoord := F6NorthingCoord + 500000;
11100
        Tr := FGNorthingCoord MOD 2000000;
11200
        Tal := Ta DIV 100000 + 1;
11300
        { Make Spheroid Adjustment }
11400
        CASE SpheroidNumber OF
11500
11600
           Clark 1866 }
11700
           1: IF (PGZoneNumber< 31) OR (PGZoneNumber > 58) THEN Tol := Tol + 10
              ELSE IF (FGZoneNumber>51) AND (FGZoneNumber<59) {take in source}
11300
11700
                          THEN Tol := Tol -10;
12000
           K International }
12100
           2: IF (PGZoneNUMBER >44) AND (PGZoneNumber <52)
12200
                  THEN Tril := Tril + 10;
12300
        {}
12400
           Clark 1880 >
12500
           3: Tril := Tril + 10;
12600
        {}
                                    30-15
```

```
12700
           { Everest >
12800
           A: IF PGZoneNumber < 46 THEN Thi := Thi + 10;
12900
        ()
13000
           { Bessel }
           5: IF (PGZoneNumber<52) AND (PGZoneLetter>/R1) THEN Toll :- Tol + 10:
13100
13200
        {}
13300
           -{ Australian }
           5: IF ODD( PGZoneNumber ) THEN Tol := Tol 15
13400
                 ELSE Tril := Tril + 5
13500
13600
        {}
13700
        END; {CASE OF SpheroidNumber}
13300
        3
13900
        Calculate Final Northins Letter Index 3
14000
        \{ \}
        IF In1 > 14 THEN In1 := In1 + 2
14100
           ELSE IF Tol > 9 THEN Tol := Tol + 1;
14200
14300
        To1 := To1 + ORD('A') - 1 +
        UTMNorthinsLetter := CHR( Tnl );
14400
14500
        { Format for Output }
       ConvertToASCII( PGNorthingCoord MOD 100000, UTMNorthingNumber );
14600
14700
       ConvertToASCII( PGEastingCoord MOD 100000; UTMEastingNumber );
14800
       UTMZoneNumber[1] := PSZoneNumber DIV 10;
14900
        UTMZoneNumber[2] := FGZoneNumber MOD 10 ;
       UTMZoneLetter := PGZoneLetter ;
15000
15100
15200
       K WRITE OUTPUT >
15300
       < >
15400
       WRITELN ;
15500
        WRITELN ;
10300
      WRITELN;
15700 . FOR I := 1 TO 2 DO WRITELN ('UTM Zune Number ',I,UTMZoneNumber[[]]);
15800
        WRITELN ('UTM Zone Letter
                                           ',UTMZoneLetter) ;
                                         ',UTMEastingLetter);
13900
        WRITELN ('UTM Easting Letter
        WRITELN ('UTM Northing Letter
16000
                                         (',UTMNorthinsLetter) ;
16100
        FOR I := 1 TO 4 DO BEGIN
16200
        WRITELN ('UTM Easting Number ',1,'
                                                 ',UTMEastingNumber[I]):
        WRITELN ('UTM Northing Number ',I,' ',UTMNorthingNumber[[]);
16300
16400
       END ;
13500
       ₹$
16600
       END OF OUTPUT >
16700
       - 60-
16800
       - END (or WITH UTM )
13700
       - < >
17000 END, (ConvertToUTM)
```

```
PROCEDURE ConvertToUTM IS
TYPE Shortarray IS ARRAY(1..4) OF CHARACTER;
                     :INTEGER: --IN
  PGNorthingCoord
   PGEastingCoord
                     :INTEGER; --IN
   PGZoneLetter
                     :CHARACTER; -- IN
  PGZoneNumber
                     :INTEGER; --IN
   SpheroidNumber
                     :INTEGER; --IN
  UTMZoneNumber
                     :ARRAY(1..2) OF INTEGER;
  UTMZoneLetter
                     :CHARACTER;
  UTMEastingLetter : CHARACTER;
  UTMNorthingLetter : CHARACTER;
   UTMEastingNumber : ARRAY(1..4) OF CHARACTER;
  UTMNorthingNumber : ARRAY(1..4) OF CHARACTER;
   I, EletterOffset, Tel, Tp, tnl, ASCIIPos: INTEGER; --Local
PROCEDURE ConvertToASCII( Number: in INTEGER;
                          AlphaNum: out ShortArray) IS
   -- Convert an integer number to its ASCII representation in base 10
   -- This procedure not detailed in source
   BEGIN
    FOR I in reverse 1 .. 4 LOOP
            ASCIIPOS := Number MOD 10 + CHARACTER'POS('0');
            AlphaNum(i) := CHARACTER'VAL( ASCIIPOS );
            Number := Number / 10;
          END LOOP;
END ConvertToASCII;
FUNCTION ODD( I: INTEGER ) return BOOLEAN is
BEGIN
   IF I = 2 * (I/2) THEN return false;
   Else return true;
   END IF;
END ODD;
BEGIN
   -- Find Easting Letter (A2)
   I >= PGZoneNumber MOD 3;
   -- Easting Letter Offet Factor
   CASE I IS
      WHEN 0=>ELetterOffet := 24;
      WHEN 1=>ELetterOffet := 6;
      WHEN 2=>ELetterOffset := 15;
      WHEN OTHERS => NULL;
   END CASE;
   Tel := ( PGEastingCoord - 500000 ) / 100000 + ELetterOffset;
   IF Tel > 14 THEN Tel := Tel + 2;
      ELSIF Tel > 9 THEN Tel := Tel + 1;
   END IF;
   Tel := Tel + CHARACTER'POS('a') - 1;
   UTMEastingLetter := CHARACTER'UAL( Tel ); -- A2
   -- Find Northing Number
   IF NOT Odd ( PGZoneNumber ) THEN
      PGNorthingCoord := PGNorthingCoord + 500000;
   END IF;
```

```
Tp := PGNorthingCoord MOD 2000000;
    Tnl := Tp / 100000 + 1;
    -- Make Spheroid Adjustment
    CASE SpheroidNumber IS
       -- Clark 1866
       WHEN 1=>IF PGGridzoneNum <31 OR PGZoneNumber> 58 THEN Tnl := Tnl + 10;
               ELSIF PGZoneNumber > 51 AND PGZoneNumber < 59 -- typo in source
                      THEN Thi := Thi - 10;
               END IF;
       -- International
       WHEN 2=>IF PGZoneNumber > 46 AND PGZoneNumber < 52 THEN Tnl := Tnl + 10;
       -- Clark 1880
       WHEN 3=>Tn1 := Tn1 + 10;
       -- Everest
       WHEN 4=>IF PGZoneNumber < 46 THEN Tnl := Tnl + 10; END IF;
       -- Bessel
       WHEN 5=>IF PGZoneNumber <52 AND PGZoneLetter <'R' THEN Tnl := Tnl + 10;
               END IF;
       -- Australian
       WHEN 6=>IF ODD( PGZoneNumber ) THEN Tn1 := Tn1 + 15;
             END IF;
      WHEN OTHERS => NULL;
                              -- Not in source
   END CASE;
   -- Calculate Final Northing Letter Index
   If Tnl > 14 Then Tnl := Tnl + 2;
      ELSIF Tn1 > 9 THEN Tn1 := Tn1 + 1;
   END IF;
   Tn1 := Tn1 + CHARACTER'POS('A') - 1;
   UTMNorthingLetter := CHARACTER'VAL( Tnl );
   -- format for output
   ConvertToASCII( PGNorthingCoord MOD 100000, UTMNorthingNumber );
   ConvertToASCII( PGEastingCoord MOD 100000, UTMEastingNumber );
   UTMZoneNum(1) := PGZoneNumber / 10;
   UTMZoneNum(2) := PGZoneNumber MOD 10;
   UTMZoneLetter := PGZoneeLetter;
END ConvertTouTM;
```

and the second of the second s

```
100
                                                                           3
 200
       PROGRAM DRVGZTB ( INPUT, OUTPUT );
 300
 400
       { PROGRAM DRVGZTB provides a test driver capability for testing >>
 500
       { GridZoneGeneration procedures for TRAILBLAZER.
 á00
 700
       TYPE
 800
          DegreesReal
                                         = Real;
 900
          ZoneRanse
                                         = 1..60;
                                         = 'A' . . 'Z';
1000
          Letters
       UAR
1100
1200
          Longitude
                                         : DesreesReal;
                                         : Degreeskeal;
1300
          Latitude
1400
                                         : ZoneRanse;
          GridZoneNumber
1500
          GridZoneLetter
                                         : Letters;
1600
                                                                           }
       PROCEDURE TBGridZoneGeneration
1700
1800
                  (COUT)
                            Lonsitude
                                                 : DesreesReal;
1900
                   {DUT}
                             Latitude
                                                 : DesreesReal;
2000
                   (IN ) VAR GridZoneNumber
                                                 : ZuneRanse;
2100
                   (IN ) VAR GridZoneLetter
                                                 : Letters);EXTERN;
2200
2300
       { PROCEDURE TRGridZoneGeneration models the TRAILBLAZER conversion}
2400
       { of seosraphic coordinates to Universal Transverse Mercator
                                                                          7-
2500
       { (UTM) coordinates srid zone designator number, letter.
                                                                           ን
2600
                                                                           Ŋ.
       -{
2700
          BEGIN
2800
      . ₹
                                                                           >
2900
             WRITELN (' ENTER Lonsitude');
3000
             READLN ( Lonsitude );
             WRITELN (' ENTER Latitude');
3100
3200
             READLN (Latitude );
3300
             TBGridZoneGeneration ( {OUT} Longitude, Latitude,
3400
                                      KIN } gridzonenumber,gridzoneletter);
3500
             WRITELN;
             WRITELN ('GridZoneNumber is ',GridZuneNumber );
3600
3700
             WRITELN;
3800
             WRITELN (' GridZoneLetter is ',GridZoneLetter );
                                                                           3-
3900
       ₹
4000
          END. { of PROGRAM DRVGZTB }
```

```
100
 200
       MODULE TEGZDG ( INFUT, OUTFUT );
                                                                           ን
 300
       TYPE
 400
 500
          DegreesRes1
                                         = REAL;
 600
          ZoneRanse
                                         - 1..60;
                                         = 'A' . . 'Z';
 700
          Letters
 800
                                                                           }
       PROCEDURE TBGridZoneGeneration
 900
1000
                  ( KIN )
                          Londitude
                                                  : DesheesResl;
                   (IN )
1100
                             Latitude
                                                  : BudreesReal;
                                                 : ZoneRanse;
                   COUTS VAR GridZoneNumber
1200
                   (OUT) VAR GridZoneLetter
1300
                                                 : Leiters);
1400
                                                                           }
1500
       C PROCEDURE TRGridZoneGeneration models the TRAILBLAZER conversion?
1600
       4 of decarable coordinates to Universal Transverse Hercator
1700
       { (UTM) coordinates - smid zone designator number, letter.
1800
1900
       4 Documentation used was the GP2UK subgrouped dtd. 20 feb 81
       C from the listings provided for the TRAILBLAZER system.
2000
2100
2200
       C PROCEDURE TBGridZoneGeneration is referenced by:
2300
                    PROGRAH DRVGZTB
2400
2500
       C PROCEDURE TRG/10ZoneGeneration makes no references.
2600
2700
       4 This procedure DOES NOT perform any data validity checks
2800
       4 that are not explicitly specified in the algorithm
2900
       4 description. This is to allow the absorithm features to be
3000
       C represented more clearly.
3100
       ₹
                                                                           }
3200
          TYPE
             Letters
                                         = 'A' . . 'Z';
3300
3400
             IndexRanse
                                         - 1..24;
3500
       €
                                                                           >
3600
          VAR
3700
             GridZoneLlrList
                                        : ARRAYE1..243 OF LETTERS;
3800
             GridZoneIndex
                                         : IndexRanse;
3900
       -{
                                                                           }
4000
          BEGIN
4100
       €
          Initialize allowable characters array
4200
4300
             GridZoneLtrList [1]
                                         $='A'$
4400
                                         :='B';
             GridZoneLtrList [2]
4500
             GridZoneLtrList [3]
                                         1='C';
4600
             GridZoneLtrList E43
                                         1 - 'B';
             GridZoneLtrList [5]
                                         :='E';
4700
             GridZoneLtrList [6]
                                         : - 'F';
4800
4900
             GridZoneLtrList [7]
                                         :='G';
             GridZoneLtrList [8]
5000
                                         1 -- 'H';
5100
             GridZoneLtrList [9]
                                         :='J';
5200
             GridZoneLtrList [10]
                                         1 = 1K 1 #
                                         1-'L';
             GridZoneLtrList [11]
5300
5400
             GridZoneLtrList [12]
                                         1-14/
5500
             BridZoneLirList [13]
                                         :='N';
5600
             GridZoneLtrList C143
                                         1-'P';
5700
             GridZoneLtrList [15]
                                         :='0';
5800
             GridZoneLtrList [16]
                                         1='R';
             GridZoneLtrList [17]
5900
                                         :='S';
             GridZoneLtrList [18]
                                         1='T';
6000
6100
             GridZumeLtrList [19]
                                         1-1011
6200
             GridZoneLtrList [20]
                                         1-1013
6300
             GridZoneLirtist [21]
                                         ; = 'W';
```

The state of the state of

and the state of

6400		GridZoneLtrList	[22]	:-'X'∮	
6500		GridZuneLtrList	0233	\$='Y';	
6600		GridZoneLtrList	E243	; - 'Z';	
6700	-{				>
0086		GridZoneNumber	:= TRUNC	(31.0+(Lonsitude/6.0));	
6900		GridZoneIndex	:- TRUNC	(13.0+(Latitude/8.0));	
7000		GridZoneLetter	:= GridZe	meLtrListCGridZoneIndex1;	
7100	₹				>
7200	END;	C of PROCEDURE TE	SeridZane6	Seneration	ን
7300	{				}
7400	FND.	4 of MODULE GROZI	r R		7

```
100
                                                                         7
 200
       PROGRAM DRVGZMG ( INPUT, DUTPUT );
 300
 400
       C PROGRAM DRVGZMG Provides a test driver carability for testing 3
       C GridZoneBeneration Procedures for MAGIIC.
 500
 600
                                                                         >
       TYPE
700
800
          Radians
                                        = REALI
900
          ZoneRanse
                                        - 1..60;
1000
          Letters
                                        = 'A'..'Z';
1100
       VAR
                                        : Radians;
          Lonsitude
1200
                                        : Radiansi
1300
          Latitude
                                        : ZoneRanse;
          GridZoneNumber
1400
                                        : Letters;
1500
          GridZoneLetter
1600
                                                                         }
       PROCEDURE MGGridZoneGeneration
1700
                 (COUT)
                                                : Radians;
1500
                            Lonsitude
1900
                  CTUO>
                            Latitude
                                                : Radians;
2000
                  (IN ) VAR GridZoneNumber
                                                : ZuneRanse;
2100
                  CIN > VAR GridZoneLatter
                                                2200
       ₹
2300
       C PROCEDURE M60ridZoneGeneration models the MA6IIC conversion
       K of seosraphic coordinates to Universal Transverse Mercator
2400
2500
       K (UTM) coordinates grid zone designator number 2 letter.
                                                                         ን
2600
                                                                         }
       ₹
2700
          BEGIN
2300
       ₹
                                                                         }
2900
             WRITELN (' ENTER Longitude');
3000
             READLN ( Lonsitude );
             WRITELN (' ENTER Latitude');
3100
3200
             READLN (Latitude );
3300
             MGGridZoneGeneration ( <COUT) Longitude, Latitude,
3400
                                     KIN > GridZoneNumber:GridZoneLetter);
3500
             WRITELN;
3600
             WRITELN (' GridZoneNumber is 'rGridZoneNumber );
3700
             WRITELN;
3800
             WRITELN (' GridZoneLetter is 'rGridZoneLetter );
3900
                                                                         }
       ₹
4000
          END. { of PROGRAM DRVGZMG }
```

```
100
       MODULE MGGZDG ( INPUT, OUTPUT );
 200
 300
       TYPE
 400
 500
          Radians
                                         = REAL;
 300
          ZoneRanse
                                         = 1..60;
 700
          Letters
                                         = 'A'..'Z';
                                                                           3
 300
 900
       PROCEDURE MGGridZoneGeneration
1000
                  (CIN )
                             Lonsitude
                                                  : Radians;
                   CIN >
1100
                             Latitude
                                                   Radiansi
                   {OUT} VAR GridZoneNumber
1200
                                                  : ZoneRanse;
                   (DUT) VAR GridZoneLetter
1300
                                                  : Letters);
1400
1500
       K PROCEDURE MGGridZoneGeneration models the MAGIIC conversion
       f of seographic coordinates to Universal Transverse Mercator
1400
1700
       { (UTM) coordinates - grid zone designator number & letter.
1300
1900
       -(
         Documentation used was source code listings from the MAGIIC
2000
       { document C6108100A dtd. 23 Oct 1978,par,3.2.90, pg.167.
2100
2200
       ₹
         PROCEDURE MCGridZoneGeneration is referenced by:
2300
                    PROGRAM DRVGZMG
2400
       ₹
2500
       4 FROCEDURE MGGridZoneGeneration makes no references.
2600
2700
       3 This procedure DOES NOT perform and data validity checks
2800
       ۲.
         that are not explicitly specified in the alsorithm
         description. This is to allow the algorithm features to be
2700
3000
       { represented more clearly.
3100
       ₹
                                                                           >
3200
          CONST
             Fi
3300
                                         = 3.1415926;
3400
          TYPE
3500
3600
             IndexRanse
                                         = 1..26;
3700
3800
          VAR
3700
             GridZoneIndex
                                        : IndexRanse;
4000
          BEGIN
4100
4200
4300
       { Calculate the grid zone number
4400
4500

« STATED LONGITUDE RANGE IS -180<-LONGITUDE<=180 IN DEGREES
</p>
4600
       ₹
4700
              GridZuneNumber := TRUNC(((130.0/Fi)*Lonsitude+130.0)/6.0)+1;
4800
4900
       { NO compensation for wrap around of srid zone numbers
5000
3100
       { Determine the srid zone letter
5200
5300

€ Since no details are provided in the referenced documentation >

5400

    ⟨ it is ASSUMED that we know how to assign A or B for latitudes ⟩

3500
       { equal to or over 84 degrees North and Y or Z for latitudes
       { equal to or over 80 degrees South.
5600
5700
5800
       4 TRUNCATION to integer is ASSUMED since it is necessary at this?
5700
       { soint in order to use the GridZoneIndex as a sointer.
6000
                                                                           `}
6100

    STATED LATITUDE RANGE 13,89
    S=LATITUDE< -84 IN DEGREES
</p>
                                                                           }
```

```
6300
              GridZoneIndex := TRUNC (((180.0/Fi)*Latitude+80.0)/8.0);
                                                                            ን
 6400
        Compute midrange grid zone letters
 6500
 6600
 5700
               IF GridZoneIndex <= 5
                  THEN GridZoneLetter := CHR(GridZoneIndex + ORD('C'));
 6800
        { Here we handle the 'i' which is not used
 5700
 7000
 7100
               IF ( GridZoneIndex >= 6 ) AND
                  ( GridZoneIndex <= 10 )
 7200
                  THEN GridZoneLetter := CHR(GridZoneIndex + ORD('C')+1);
 7300
 7400
        { Here we handle the 'O' which is not used
                                                                            <u>٠</u>
 7500
                                                                            }-
 7300
        ί.
 7700
               IF ( GridZoneIndex >= 11 ) AND
 7800
                  ( GridZoneIndex <= 19 )
 7900
                  THEN GridZoneLetter := CHR(GridZoneIndex + ORD('C')+2);
 3000
        ₹
        { The rest of the GridZoneIndex are biased off by ORD ('C')
                                                                            3.
 3100
                                                                            }
 8200
        ₹
               IF GridZoneIndex > 19
 3300
                  THEN GridZoneLetter := CHR(GridZoneIndex);
 8400
 3500
        { Assign Y or Z to the North Polar Zone according as Western
 3600
        { or Eastern Hemisphere, respectively.
 3700
                                                                            >
 8800
        ₹.
3700
              IF Latitude*(180.0/Pi) >= 84.0
 9000
                  THEN IF Londitude*(180.0/Pi) < 0.0
 9100
                          THEN GridZoneLetter := 'Y'
                          ELSE GridZoneLetter := 'Z';
 9200
 9300
                                                                            >
 9400
        { Assign A or B to the South Polar Zone according as Western
 9500
        { or Eastern Hemisphere, respectively.
                                                                            }·
9600
 9700
              IF Latitude*(180.0/Pi) <= -80.0</pre>
9800
                  THEN IF Longitude*(180.0/Pi) < 0.0
 9900
                          THEN GridZoneLetter := 'A'
10000
                          ELSE GridZoneLetter := 'B';
10100
        ₹
                                                                            }-
10200
        { NO correction for the four irresular zones -
             32X,34X, and 36X do not exist
10300
        ₹
10400
             31V is truncated
        ₹
10500
        {
10600
        END; { of PROCEDURE MGGriDZoneGeneration
10700
10800
        END. { of MODULE MGGZDG
```

0244

```
100
                                                                            ን
 200
       PROGRAM DRVGZBT ( INPUT, OUTPUT );
 300
 400
       K PROGRAM DRVGZBT provides a test driver capability for testing >>
 500
       & GridZoneGeneration procedures for BETA.
 600
 700
       TYPE
 800
          Radians
                                         = REAL;
 900
          ZoneRanse
                                         = 1..60;
                                         = 'A'..'Z';
1000
          Letters
1100
       VAR
                                         : Radians;
1200
          Lonsitude
1300
          Latitude
                                         : Radians;
          GridZoneNumber
1400
                                         : ZoneRanse;
          GridZoneLetter
1500
                                         : Letters;
1600
          CenterMeridian
                                         : Radians;
                                                                            }
1700
1800
       PROCEDURE BTGridZoneGeneration
1900
                  ({OUT}
                             Longitude
                                                  : Radians;
2000
                   (OUT)
                             Latitude
                                                  : Radians;
2100
                   (IN > VAR GridZoneNumber
                                                  : ZoneRanse;
2200
                   {IN } VAR GridZoneLetter
                                                  : Letters;
2300
                   (IN ) VAR CenterMeridian
                                                  : Radians);EXTERN;
2400
2500

    FROCEDURE BTGridZoneGeneration models the BETA conversion

       ₹ of seographic coordinates to Universal Transverse Mercator
2600
2700
       { (UTM) coordinates srid zone designator number, letter and the
2800
       { central meridian of the rectangle.
                                                                           ን
2700
       ₹
                                                                            }
3000
          BEGIN
3100
       ₹
                                                                           ን
3200
              WRITELN (' ENTER Lonsitude');
3300
             READLN ( Longitude );
3400
             WRITELN (' ENTER Latitude');
3500
             READLN (Latitude );
3600
             BTGridZoneGeneration ( {OUT} Longitude, Latitude,
3700
                                      {IN } GridZoneNumber,GridZoneLetter,
3800
                                             CenterMeridian);
3900
             WRITELN;
4000
             WRITELN (' GridZoneNumber is ',GridZoneNumber );
4100
             WRITELN;
4200
             WRITELN (' GridZoneLetter is ',GridZoneLetter );
4300
             WRITELN;
4400
             WRITELN (' CenterMeridian is ',CenterMeridian )
                                                                           Э.
4500
4600
          END. ( of PROGRAM DRVGZBT )
```

```
100
                                                                           >
 200
       MODULE BIGING ( INPUT, OUTPUT );
 300
 400
       TYPE
          Radians
 500
                                         = REAL;
 600
          ZoneRanse
                                         - 1..60;
700
          Letters
                                         = 'A' . . 'Z';
800
900
       PROCEDURE BTGridZoneGeneration
1000
                  ( KIN )
                             Lungitude
                                                  : Radiansi
1100
                   < NI>
                             Latitude
                                                  : Radiansi
                   COUTY VAR GridZoneNumber
                                                  : ZoneRanset
1200
                   {OUT} VAR GridZoneLetter
                                                 : Letters!
1300
1400
                   COUTS VAR CenterHeridian
                                                  : Radiana);
1500
                                                                           }
1600
       C PROCEDURE BTGridZoneGeneration models the BETA conversion
1700
       { of seographic coordinates to Universal Transverse Mercator
1800
       C (UTM) coordinates - said zone designator number, letter, and
1900
       { the central meridian.
2000
       ₹
                                                                           3.
2100
       C Documentation used was source code listings from the RETA
                                                                           >
       { document 8822-43 dtd. 16 Oct 1981; apx.4;ps.2-474 for the
2200
2300
       C ADSCNU subprogram and pg.2-450 for the ADSCCM subprogram
2400
2500
       K PROCEDURE BIGridZoneGeneration is referenced by:
2600
                    PROGRAM DRYGZBT
       ₹.
2700
2800
       C PROCEDURE BIGridZoneGeneration makes no references.
2900
                                                                           3
       f This procedure DOES NOT perform and data validity checks
3000
       I that are not explicitly specified in the algorithm
3100
3200
       4 description. This is to allow the alsorithm features to be
3300
       4 represented move clearly.
3400

    Since the included 'ZDBFRO.COM' is not available to us at

3500
       K this time, we assume implicit typing in the source FORTRAN
3600
3700
       < code.
3800
3900
          CONST
4000
             Fi
                                         = 3.1415726;
4100
                                                                           3.
          TYPE
4200
                                         = 'A'..'Z';
4300
             Letters
4400
             IndexRanse
                                         -- 1..24;
4500
                                                                           }
4600
          VAR
                                        : ARRAYCI..243 OF LETTERS;
4700
             GridZoneLtrList
4800
             GridZoneIndex
                                         : IndexRandet
4900
                                                                           >
       ₹
5000
          BEGIN
5100
       ₹
          Initialize allowable characters array
                                                                           >
5200
5300
             GridZoneLtrList [1]
                                         :='A';
                                         :-'B';
             GridZoneLtrList [2]
5400
                                         :='C';
5500
             GridZoneLirList [3]
                                         : " (1) " ;
             GridZonettrList [43
5600
                                         1='E';
5700
             GridZoneLtrList [5]
                                         :-'F';
             GridZoneLtrList [6]
5800
             GridZoneLtrList [7]
                                         :='G';
5900
             GridZonettrList [8]
                                         1-'H';
6000
                                         :='(1';
6100
             GridZoneLirList [7]
                                         1-1K19
6200
             GridZoneLtrList [10]
                                         :='L';
6300
             GridZoneLittist [11]
                                   20-26
```

```
6400
              GridZonettrtist [12]
                                          $ -- ' 治 ' 身
                                          :='%';
6500
              GridZoneLirtist [13]
                                          :='P';
6600
              GridZoneLtrList [14]
6700
              GridZoneLirList [15]
                                          :='@';
6800
              GridZoneLtrList [16]
6700
              GridZoneLtrList [17]
7000
              GridZoneLtrList [18]
7100
              GridZoneLtrList [19]
                                          := '11';
                                          : ·'V';
7200
              GridZunettrList [20]
              GridZoneLtrList [21]
7300
                                          :='W';
7400
              GridZoneLtrList [22]
                                          : - 'X';
                                          ;='Y';
7500
              GridZoneLtrList [23]
7600
                                          :='Z';
              GridZoneLirList [24]
7700
7800
        Calculate the drid zone number
7900
8000
              GridZoneNumber := (TRUNC(((18000.0/Fi)%Lonsibude)
8100
                                             +18600.0)) DIV 600;
8200
8300
        K Compensate for wram around of Brid zone numbers
8400
8500
              IF GridZuneNumber > 60
                 THEN GridZoneNumber := GridZoneNumber-60;
8600
8700
              IF GridZuneRumber < 1
8800
                 THEN GridZoneNumber := GridZoneNumber+60;
8900
9000
        C Determine the drid zone letter
9100
9200
              GridZoneIndex := TRUNC((Latitude*(180.0/Fi)+194.0)/8.0);
9300
9400
        4 Test for and lock out North Polar Zones
                                                                            }
9500
9600
              IF GridZoneIndex > 22
9700
9800
                 THEN GridZoneIndex :- 22;
9900
10000
        C NOTE that no such test is needed for the South Polar Zone
10100
        C because the algoritm limit was given as <= 80 South</p>
10200
                                                                            ን
        ₹
10300
              GridZoneLetter := GridZoneLtrListfGridZoneIndex1;
10400
10500
          Correct for the four irresular zones -
10600
             32X+34X+ and 36X do not exist
        ₹
10700
        ₹
             31V is truncated
10800
10700
        4 Truncate Grid Zone 31V
11000
11100
              IF (GridZoneIndex - 20) AND
11200
                  ((GridZoneRumber =31) AND
                  (Lonstitude >= 3.0*(Fi/180.0)))
11300
11400
                  THEN GridZoneRomber : 32;
11500
        Correct for smid zones 32X,34X, and 36X
11600
11700
11800
              IF (GridZuneIndex =22) AND
11900
                  (GridZoneNumber ~32)
                  THEN IF Lungitude >= 9.0*(Pi/180.0)
12000
12100
                          THEN GridZoneNumber := 33
                          ELSE GridZoneNumber : 31;
12200
                                                                            >
12300
12400
12500
              IF (BridZuneIndex - 22) AND
                  (BridZoneNumber - 34)
12600
```

```
12700
                 THEN IF Lungitude >- 21.0%(Pi/180.0)
12800
12900
                         THEN GridZoneNumber : 35
13000
                         ELSE GridZoneNumber :- 33;
13100
                                                                          ን
              IF (GridZoneIndex -22) AND
13200
13300
                 (GridZuneNumber - 36)
                 THEN XF Longitude >= 33.0*(Pi/180.0)
13400
13500
                         THEN GridZoneNumber :- 37
13600
                          ELSE GridZoneNumber := 35#
13700
       -{
                                                                          ን
              CenterNeridian :- (6*GridZoneNumber-183)*(Pi/180.0)
13800
13900
                                                                          }
14000
        end; { of PROCEDURE BIGriDZoneGeneration
14100
14200
        END. C of MODULE BIGZDG
```

```
100
       PROGRAM DRVGZGR ( INPUT, OUTFUT );
 200
300
 400
       C PROGRAM DRVGZGR provides a test driver capability for testing >
500
       C GridZoneGeneration Procedures for GUARDRAIL.
700
                                                                          }
800
       TYPE
900
          DesreesReal
                                        = Real ;
1000
                                        - INTEGER;
          DesreesInteser
1100
          ZoneRanse
                                        = 1..60;
1200
          Letiers
                                        = 'A'..'Z';
1300
       VAR
1400
          Longitude
                                        : DegreesRoal;
1500
          Latitude
                                        : DesreesReal;
          GridZoneNumber
1600
                                        : ZuneRansei
1700
          GridZoneLetter
                                        : Lotters;
1750
          centermeridian
                                        : desireesintesier;
                                                                         γ.
1800
1900
       PROCEDURE GROwidZoneGeneration
                                                 : DedreesReal;
2000
                 <TU03>
                            Lundibude
                                                 : DesreesReal;
2100
                  CTUO3
                             Latitude
                  (IN ) VAR GridZoneNumber
                                                 : ZoneRansei
2200
                  (IN ) VAR GridZoneLetter
                                                 : Letters;
2300
                  (IN > VAR CenterMoridian
2400
                                                 2500
2603
       C PROCEDURE GRanidZoneGeneration models the GUARDRAIL conversion?
       C of seosraphic coordinates to Universal Transverse Mercator
2604
2606
       (UTA) coordinates grid zone designator number/letter and the
2608
         central meridian of the rectangle.
2900
3200
          BEGIN
3300
                                                                          ን
       ₹
             WRITELM ( 'ENTER Longitude');
3400
3500
             READLN ( Lonsitude );
3600
             WRITELN (' ENTER Latitude');
3700
             READLN (Latitude );
3800
             GRGridZoneGeneration ( {OUT} Longitude:Latitude:
3900
                                     KIN } gridzonenombersgridzonelottors
3950
                                            centermeridian);
4000
             WRITELN;
4100
             WRITELN (' GridZoneRomber is 'rGridZoneRomber );
4200
             WRITELN;
4300
             WRITELN (' GridZoneLetter is ':GridZoneLetter );
4400
             WRITELNI
4500
             WRITELN (' CenterMeridian is '*ContorMeridian )
4550
                                                                          3
4600
          END. { of PROGRAM DRVGZGR }
```

```
100
                                                                            ን
 200
       MODULE GRGZDG ( INPUT, OUTPUT );
 300
                                                                            3
 400
       TYPE
 500
          DesreesReal
                                          = REAL;
 600
          DesressInteser
                                          - INTEGER;
 700
          ZoneRanse
                                          = 1..60;
 800
          Letters
                                          - 'A' . . 'Z';
 900
                                                                            }-
1000
       PROCEDURE GRGeidZoneGeneration
1100
                  Longitude
                                                  : DesreesReal;
1200
                   { NI}
                             Latitude
                                                  : DesreesRes1;
1300
                   COUTS VAR GridZoneNumber
                                                  : ZuneRansie)
1400
                   COUTY VAR GridZoneLetter
                                                  : Lobbers;
1500
                   COUTS VAR CenterMeridian
                                                  : DesreesInteser);
1600
1700
       C PROCEDURE GRGridZoneGeneration models the GUARDRAIL conversion?
       C of geographic coordinates to Universal Transverse Mercator
1800
1900
       4 (UTM) coordinates - smid zone designator numbers letters and
2000
       C the central meridian.
2100
2200
       C PROCEDURE GRG ridZoneGeneration is referenced by:
2300
                    PROGRAM DRUGZGR
2400
2500
       4 PROCEDURE GROwidZoneGeneration makes no references.
2600
2700
       { This procedure DOES NOT perform any data validity checks
       C that are not explicitly specified in the algorithm
2800
2900
       4 description. This is to allow the alsorithm features to be
3000
       C represented more clearly.
3100
                                                                            }
3200
          TYPE
3300
             Letters
                                         = 'A' . . 'Z';
3400
             IndexRanse
                                         - 1..24;
3500
                                                                            >
3600
          VAR
3700
             GridZoneLirList
                                         : ARRAYE1..243 OF LETTERS;
              GridZoneIndex
3800
                                          : IndexRansiei
3900
       €
                                                                            }
4000
4100
          Initialize allowable characters arras
4200
4300
              GridZoneLtrList [13
                                          1='6';
4400
              GridZoneLtrList [2]
                                          : - 'B';
4500
              GridZoneLtrList [3]
                                         :='C';
4600
              GridZonettrList [4]
                                          1-10/#
4700
              GridZoneLtrList [5]
                                          1-'E';
                                          :-'F':
              GridZoneltrList [6]
4800
4900
              GridZoneLtrList [7]
                                          1 = 'G' +
              GridZoneLtrList [8]
5000
                                          1 - 'H';
              GridZoneLtrList [9]
                                          :='1';
5100
5200
                                          :-'K';
              GridZoneLtrList [10]
              GridZoneLtrList [11]
                                          :='L';
5300
              GridZonet.trList [12]
                                          : - 'M' ;
5400
5500
              GridZoneLtrList [13]
                                         :='R';
5600
              GridZoneLtrList C143
                                          1 - 'P';
5700
              GridZoneLtrList [15]
                                         :='0';
                                          :-- 'R';
5800
              BridZoneLtrList [16]
5900
              GridZoneLivLivt [17]
                                         1=131
6000
              GridZoneLtrList [13]
                                          1 - 'T';
6100
              GridZuneLtrList [19]
                                          :='U';
              BridZoneLtrList [20]
                                          : ''';
6200
              BridZuneLinkist [21] 30-30
6300
                                         :='\';
```

المرازاء ويوريها يومانها فلاعوا يوواي الد

California de la companya del companya del companya de la companya

```
6400
                 GridZonettrList [22]
                                                     1:-- 'X ' j
                                                     ; = 'Y';
6500
                 GridZuneLtrList [23]
6600
                 GridZoneLtrList 6243
6700
                                                                                                  >
                 GridZoneNumber := TRUNC (31.0+(Longitude/6.0));
GridZoneIndex := TRUNC (13.0+(Longitude/6.0));
GridZoneLetter := GridZoneLtgtistEGridZoneIndex1;
6800
6900
70001
7100
                 CenterMeridian :- 6#GridZoneNumber-183
7200
         ₹
                                                                                                  ን
7300
         END; C of PROCEDURE GRGriDZoneGeneration
                                                                                                             >
7400
                                                                                                  ን
         END. { of MODULE GRGZDG
7500
```

FLED

DTIC

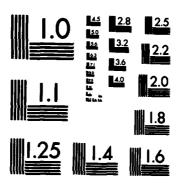


AD-A129 182

US ARMY INTELLIGENCE CENTER AND SCHOOL USAICS SOFTWARE ANALYSIS AND MANAG. (U) JET PROPULSION LAB PASADENA CA P BABBY ET AL. 89 JUL 82 JPL-D-181 NAS7-918

UNCLASSIFIED

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SUPPLEMENTARY

INFORMATION

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

REPORT DOCUMENTA	READ INSTRUCTIONS BEFORE COMPLETING FORM	
TALCOS D-18/	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
Analysis of Geographic Tr Algorithms	5. TYPE OF REPORT & PERIOD COVERED FINAL	
		6. PERFORMING ORG. REPORT NUMBER D=181
7. AUTHOR(s) J. Gillis, J. Radbill, A Palmer, P. Babby	8. CONTRACT OR GRANT NUMBER(*) NAS7-918	
PERFORMING ORGANIZATION NAME AND AI Jet Propulsion Laboratory California Institute of ' 4800 Oak Grove, Pasadena	10. PROGRAM ELEMENT, PROJECT, TASK REARE 82 WORK UNIT NUMBERS AMEND # 187	
1. CONTROLLING OFFICE NAME AND ADDRES COmmander, USAICS	12. REPORT DATE July 9, 1982	
ATTN: ATSI-CD-SF Ft. Huachuca, AZ 85613-	13. NUMBER OF PAGES 35	
14. MONITORING AGENCY NAME & ADDRESS(II (Same as Item 11)	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NONE

Approved for Public Dissemination

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Prepared by Jet Propulsion Laboratory for the US Army Intelligence Center and School's Combat Developer's Support Facility

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

GEOGRAPHIC TRANSFORMATION, COORDINATE TRANSFORMATION, ALGORITHMS, MAGIIC, BETA, GUARDRAIL, TRAILBLAZER, ALGORITHM ANALYSIS, SOFT-WARE ANALYSIS, IEW SYSTEMS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the findings of JPL regarding geographic transformation algorithms used in MAGIIC, GUARDRAIL, TRAILBLAZER and BETA systems. A set of parameters is developed to characterize and catalogue intelligence system algorithms in the four sys-Individual algorithms are also analyzed to determine if they are performing their functions properly.

END

FILMED

9-85

DTIC

REPRODUCED AT GOVERNMENT EXPENSE